kaspersky

Kaspersky Unified Monitoring and Analysis Platform

© 2026 AO Kaspersky Lab

Contents

Kaspersky Unified Monitoring and Analysis Platform Help
About Kaspersky Unified Monitoring and Analysis Platform
- What's new
- Distribution kit
- Hardware and software requirements
- KUMA interface
- Compatibility with other applications

Architecture of the solution
- Core
- Collector
- Correlator
- Storage
- Basic entities
  - Tenant
  - Event
  - Alert
  - Incident
  - Assets
  - Resources
  - KUMA services
  - KUMA agents
  - Severity

Licensing the solution
- About the End User License Agreement
- About the license
- About the License Certificate
- About the license key
- About the key file
- About the activation code
- Data provision in Kaspersky Unified Monitoring and Analysis Platform
- Adding a license key
- Viewing information about the added license key
- Deleting a license key

Administrator's guide
- Installing and removing KUMA
  - Solution installation requirements
    - Upgrading Oracle Linux 8.x to Oracle Linux 9.x with installed KUMA services
  - Ports used by KUMA during installation
  - Downloading CA certificates
  - Reissuing internal CA certificates
  - Modifying the self-signed web console certificate
  - Synchronizing time on servers

About the inventory file
KUMA settings in the inventory file
Installation on a single server
Preparing the single.inventory.yml inventory file
Installing KUMA on a single server
Changing the hostname
Distributed installation
Preparing the control machine
Preparing the target machine
Preparing the distributed.inventory.yml inventory file
Installing KUMA in a distributed configuration
Changing the hostname of microservices
Distributed installation with KUMA Core in a Kubernetes cluster
Additional requirements for deploying KUMA Core in Kubernetes
Installing KUMA in a Kubernetes cluster
Preparing the control machine
Preparing the target machine
Preparing the k0s.inventory.yml inventory file
Installation with KUMA Core in a Kubernetes cluster
Migrating the KUMA Core to a new Kubernetes cluster
KUMA Core availability under various scenarios
Managing Kubernetes and accessing KUMA
Time zone in a Kubernetes cluster
Maintenance of hosts in a Kubernetes cluster
Restoring a Kubernetes cluster
Constituent parts of the Kubernetes cluster and snapshots of their virtual machines
Failure scenarios
Distributed installation with KUMA Core in a Raft cluster
Preparing the control machine
Preparing the target machine
Preparing the distributed.inventory.yml file for installing KUMA with multiple KUMA Core services on a Raft cluster
Installing in a high availability configuration in a Raft cluster
Configuring the balancer for the Raft cluster
Example configuration of HAProxy Community Edition
Example configuration of NGINX Open Source
Expanding the Raft cluster
Deleting a Raft cluster node
Restoring a Raft cluster node
Restoring the Raft cluster
Restoring the Raft cluster from the current state of a node
Restoring the Raft cluster from backup
Changing the hostname of the KUMA Core
KUMA backup
Creating a backup of the KUMA Core
Restoring the KUMA Core for a single-server installation
Restoring the KUMA Core for a distributed installation

- Restoring KUMA in a Kubernetes cluster
- Restoring KUMA Core in a Raft cluster
- Modifying the configuration of KUMA
- Updating previous versions of KUMA
- Upgrading components of the Kubernetes cluster
  - Upgrading Kubernetes cluster components without the -e 'manual_restore_only=true' option
  - Upgrading Kubernetes cluster components with the -e 'manual_restore_only=true' option
- Sizing the metrics data in an installation with KUMA Core 3.4.x in a Kubernetes cluster
- Growing the KUMA Core volume or the Metrics service volume in an active Kubernetes cluster
- Shrinking the Metrics service volume
  - Shrinking a Metrics service volume without data loss
  - Shrinking a Metrics service volume with data loss
- Migrating alert events to ClickHouse after upgrading KUMA from version 3.4.x
  - Settings for alert event migration from MongoDB to ClickHouse
- Troubleshooting update errors
- Reissuing k0s cluster certificates upon expiration
- Delete KUMA

Working with tenants
- Selecting a tenant
- Tenant affiliation rules

Managing users
- User roles
- Rights of predefined roles
- Creating a user role
- Functional areas and permissions available to them
- Editing user role settings
- Viewing role settings
- Deleting a user role
- Configuring the security policy for logging in to the web interface
- Creating a user account
- Editing a user account
- Special considerations involved in changing the set of spaces assigned to a user

KUMA services
- Services tools
  - Renaming a service
  - Getting service identifier
  - Stopping, starting, checking status of the service
  - Restarting the service
  - Deleting the service
  - Viewing the storage partition table
  - Viewing the table of active queries
  - 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
  - Disconnecting, archiving, and connecting partitions
- Creating a set of resources for a storage
- Creating a storage service in the web interface
- Installing a storage in the KUMA network infrastructure
- Recommendations for using system memory
- Backing up, restoring, and importing partitions
  - Scheduling partition backup
  - Manually backing up partitions
  - Restoring partitions from backup
  - Importing partitions from an external KUMA installation

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 the correlator on the server
- Validating correlator installation
- Creating a correlator-ng correlator
  - Configuring correlator-ng
  - Deploying correlator-ng in a Raft cluster
  - Supported data types for correlator-ng
  - Correlation rule selectors and expressions
  - Attributes of the standard correlation rule
  - Data processing 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. Transport
  - 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
  - Managing connections for an agent
  - Creating an agent service in the KUMA web interface
  - Automatically created agents
  - Installing an agent in a KUMA network infrastructure
    - Installing a KUMA agent on Linux assets
    - Installing a KUMA agent on Windows assets
  - 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 computers to KUMA
  - Importing a file with a list of servers for WMI agents
- AI services
  - AI asset score and status
    - Installing and removing the AI asset score and status service
    - Reissuing the AI service certificate
    - Settings of the AI asset score and status service
  - Analyze with KIRA
    - Configuring integration with KIRA
    - Analyzing using KIRA
    - Possible errors of the Analyze with KIRA task
  - Configuring DLL Hijacking detection
- Metrics service
  - Installing the Metrics service on a host
- Configuring event sources
  - Configuring receipt of Auditd events
    - Configuring the KUMA collector for receiving Auditd events
    - Configuring the event source server using the rsyslog service
    - Configuring the event source server using the syslog-ng service
  - 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 the export of Kaspersky Security Center events to the KUMA SIEM system
    - Configuring export of Kaspersky Security Center events in CEF format
    - Configuring KUMA collector for collecting Kaspersky Security Center events
    - Installing KUMA collector for obtaining Kaspersky Security Center events

Configuring receiving Kaspersky Security Center 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 DNS server events using the ETW agent
Configuration on the Windows side
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 the receipt of KICS for Networks events
Creating a KICS for Networks connector for sending events to KUMA
Configuring export of KICS for Networks events to KUMA
Creating a KUMA collector to receive KICS for Networks events

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 audit 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 Windows event reception using Kaspersky Endpoint Security for Windows
Configuring Linux event reception using Kaspersky Endpoint Security for Linux
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

- Configuring receipt of VK WorkSpace Mail events
  - Configuring the export of VK WorkSpace Mail events
- Configuring receipt of Angie events
  - Configuring the export of Angie events
- Configuring receipt of Dallas Lock 8 events
  - Configuring the export of Dallas Lock 8 events
- Configuring receipt of Dallas Lock Unified Control Center events
  - Configuring the export of Dallas Lock UCC events
- Configuring receipt of ProFTPD events
  - Configuring audit of ProFTPD events.
  - Configuring a syslog server to send ProFTPD events
- Configuring receipt of Vsftpd events
  - Configuring audit of Vsftpd events.
  - Configuring a syslog server to send Vsftpd events
- Configuring receipt of HAProxy events
  - Configuring export of HAProxy events
- Configuring receipt of Veeam Backup & Replication events
  - Configuring the export of Veeam Backup & Replication events
- Configuring receipt of Apache Tomcat events
  - Configuring Apache Tomcat event audit
  - Configuring the syslog server to send Apache Tomcat events.

Monitoring event sources

- Source status
  - List of event sources
- Monitoring policies

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 the KUMA web interface
  - Importing asset information and asset vulnerability information from Kaspersky Security Center
  - Importing asset information from MaxPatrol
    - Importing data from MaxPatrol reports
    - Importing asset information from MaxPatrol VM
      - Settings of the kuma-ptvm-config.yaml configuration file
  - Importing asset information from KICS for Networks
  - Examples of asset field comparison during import
- 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
- Detection of unknown assets
- Custom asset fields
- Critical information infrastructure assets

Integration with other solutions

- Integration with Kaspersky Security Center
  - Configuring Kaspersky Security Center integration settings
  - Adding a tenant to the list for Kaspersky Security Center integration
  - Removing a tenant from the list of tenants for Kaspersky Security Center integration
  - Creating Kaspersky Security Center connection
  - Editing Kaspersky Security Center connection
  - Deleting Kaspersky Security Center connection
  - 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 KUMA event details
- Integration with Kaspersky CyberTrace
  - Integrating CyberTrace indicator search
    - Configuring CyberTrace to receive and process requests
    - Creating event Enrichment rules
  - Integrating CyberTrace interface
  - Configuring event enrichment and retroscan
- Integration with Kaspersky Threat Intelligence Portal
  - Initializing the Kaspersky Threat Intelligence Portal integration
  - Requesting information from Kaspersky Threat Intelligence Portal
  - Viewing information from Kaspersky Threat Intelligence Portal
  - Updating information from Kaspersky Threat Intelligence Portal
- Integration with R-Vision Security Orchestration, Automation and Response
  - Configuring integration in KUMA
  - Configuring integration in R-Vision SOAR
    - Adding the ALERT_ID and ALERT_URL incident fields
    - Creating a collector in R-Vision SOAR
    - Creating connector in R-Vision SOAR
    - Creating rule for closing KUMA alert when R-Vision SOAR incident is closed
  - Managing alerts using R-Vision SOAR
- Integration with Active Directory, Active Directory Federation Services and FreeIPA
  - 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
Authentication using domain accounts
Enabling and disabling domain authentication
Configuring connection between KUMA and FreeIPA
Configuring connection between KUMA and Active Directory
Configuring connection between KUMA and Active Directory Federation Services
Configuring connection in Active Directory Federation Services
Fixing the 'Access denied' error when logging in to KUMA via ADFS
NCIRCC integration
Integration with the Security Orchestration Automation and Response Platform (SOAR)
Configuring integration in KUMA
Configuring integration in SOAR
Importing and configuring a connector
Configuring the handler, schedule, and worker process
Integration with KICS for Networks and KATA
Configuring integration in KICS for Networks or KATA
Configuring integration in KUMA
Enabling and disabling integration with KICS for Networks or KATA
Changing the data update frequency
Special considerations when importing asset information from KICS for Networks or KATA
Changing the status of a KICS for Networks or KATA asset
Kaspersky Automated Security Awareness Platform
Creating a token in ASAP and getting a link for API requests
Configuring integration in KUMA
Viewing information about the users from ASAP 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
Configuring a response rule for integration with KSMG
Kaspersky Managed Detection and Response integration
Preliminary configuration
Initializing integration
Importing asset information from RedCheck

- Configuring receipt of Sendmail events
  - Configuring Sendmail logging
  - Configuring export of Sendmail events
- Managing KUMA
  - Logging in to the solution web interface
  - Configuring the KUMA Core
  - Service monitoring
    - Viewing KUMA metrics
    - KUMA metric alert triggering conditions
  - Managing KUMA tasks
    - Viewing the tasks table
    - Configuring the display of the tasks table
    - Viewing task run results
    - Restarting a task
  - 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 notifications
  - KUMA logs
- Working with geographic data
  - Geodata format
  - Converting geographic data from MaxMind to IP2Location
  - Importing and exporting geographic data
  - Default mapping of geographic data

User guide

- KUMA resources
  - Operations with resources
    - Creating, renaming, moving, and deleting resource folders
    - Creating, duplicating, moving, editing, and deleting resources
    - Bulk deletion of resources
    - Linking resources to services
    - Updating resources
      - Configuring a custom source using Kaspersky Update Utility
    - Exporting resources
    - Importing resources
    - Resource search
    - Tag management
    - Resource usage tracing
    - Resource versioning
      - Comparing resource versions
      - Restoring a resource version
      - Configuring the retention period for resource versions

Destinations
Destination, internal type
Destination, type nats-jetstream
Destination, tcp type
Destination, udp type
Destination, http type
Destination, diode type
Destination, kafka type
Destination, file type
Destination, storage type
Destination, 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 sending data to an extra normalizer
Supported event sources
Aggregation rules
Enrichment rules
Configuring the enrichment of collectors with assets
Applying an enrichment rule when managing collector settings
Data collection and analysis rules
Configuring the scheduler for a data collection and analysis rule
Correlation rules
Correlation rules of the 'standard' type
Correlation rules of the 'simple' type
Correlation rules of the 'operational' type
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
Adding a temporary exclusion list for a correlation rule

- Predefined correlation rules
- MITRE ATT&CK techniques and tactics
  - Adding MITRE enrichment rules to the correlator
  - Importing the list of MITRE ATT&CK techniques
  - Mapping MITRE ATT&CK techniques to correlation rules
  - Exporting correlation rules to MITRE ATT&CK Navigator
- Sigma rules converter
  - Preparing a Sigma rule
  - Examples of supported Sigma rule conversions

Filters

Active lists

- Viewing the table of correlator active lists
- Adding active list
- Viewing the settings of an active list
- Changing the settings of an active list
- Duplicating an active list
- Deleting active lists
- 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 entries to an active list
- Exporting records from the active list
- Predefined active lists

Proxies

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
- Response rules with EDR actions
  - Configuring response rules with EDR actions

Notification templates

- Creating a notification template
- Editing a notification template
- Deleting a notification template
- Notification template syntax

Connectors

- Viewing connector settings
- Adding a connector
- Connector settings
  - Connector, internal type
  - Connector, tcp type

Connector, udp type
Connector, netflow type
Connector, sflow type
Connector, nats-jetstream type
Connector, kafka type
Connector, http type
Connector, sql type
Connector, file type
Connector, 1c-log type
Connector, 1c-xml type
Connector, diode type
Connector, ftp type
Connector, nfs type
Connector, wmi type
Connector, wec type
Connector, etw type
Connector, snmp type
Connector, 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
Connector, kata/edr type
Connector, vmware type
Connector, elastic type
Connector, office365 type
Connector, dfi type
Connector, universalAPI type
Predefined connectors
Secrets
Segmentation rules
Creating a segmentation rule
Linking segmentation rules to correlation rules
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 and finding records in the context table
Adding, editing, and deleting records in the context table
Exporting data from a context table
Importing data into a context table
Example of incident investigation using Kaspersky SIEM
Incident conditions
Step 1. Preliminary steps
Step 2. Assigning an alert to a user

- Step 3. Check if the triggered correlation rule matches the data of the alert events
- Step 4. Analyzing alert information
- Step 5. False positive check
- Step 6. Determining alert severity
- Step 7. Incident creation
- Step 8. Investigation
- Step 9. Searching for related assets
- Step 10. Searching for related events
- Step 11. Recording the causes of the incident
- Step 12. Incident response
- Step 13. Restoring assets operability
- Step 14. Closing the incident

Analytics

- Working with events
  - Filtering and searching events
    - Selecting Storage
    - Generating an SQL query using a builder
    - Manually creating an SQL query
    - Saving query history
    - Managing saved search queries
    - Filtering events by period
    - Displaying the list of events with a relative and absolute calendar period
    - Grouping events
    - Displaying names instead of IDs
    - Presets
    - Limiting the complexity of queries in alert investigation mode
    - Saving and selecting events filter configuration
    - Deleting event filter configurations
    - Supported ClickHouse functions
    - Viewing event detail areas
    - Exporting events
    - Configuring the table of events
    - Refreshing events table
    - Getting events table statistics
    - Viewing correlation event details
    - Generating an SQL query using KUMA SQL functions
      - Using the 'enrich' function set in SQL queries
      - Using the 'lookup' function set in SQL queries
  - Event route tracing
  - Categorization of events
  - Granular access to events
- Dashboards
  - Creating a dashboard
  - Selecting a dashboard
  - Selecting a dashboard as the default dashboard
  - Managing dashboard settings
  - Deleting a dashboard

- Enabling and disabling the TV mode
- Setting the period without modifying dashboard settings
- Predefined dashboards
- Universal dashboards
- Exporting dashboards
- Importing dashboards

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

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 custom widget
  - Image widget
  - Text widget
  - Other widgets
- Filtering widget data
  - Filtering data on a single widget
  - Filtering data in multiple widgets
  - Drilling down to a related dashboard
- Displaying tenant names in widgets of the "Active list" type

Working with alerts
- Configuring alerts table
  - Filtering alerts
  - Saving and selecting an alert filter
  - Deleting an alert filter
- Viewing details on an alert
- Changing alert name

Configuring alert filling
Processing alerts
Alert investigation
Retention of alerts and incidents
Alert notifications
Working with incidents
About the incidents table
Saving and selecting incident filter configuration
Deleting incident filter configurations
Viewing information about an incident
Incident creation
Incident processing
Changing incidents
Automatic linking of alerts to incidents
Categories and types of incidents
Interaction with NCIRCC
Exporting data to NCIRCC
Getting incidents from NCIRCC
Supplementing incident data on request
Sending files to NCIRCC
Communication with NCIRCC experts
Supported categories and types of NCIRCC incidents
Notifications about the incident status change in NCIRCC
Retroscan
Contacting Technical Support
REST API
Creating a token
Configuring permissions to access the API
Authorizing API requests
Standard error
REST API v3 operations
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
Alert data model
Asset data model
User account data model
KUMA audit events
User was successfully signed in or failed to sign in
User login changed
Other data of the user was changed
User logged out
User password was changed
User was created
User role was assigned

User role was revoked

The user has edited the set of fields settings to define sources

Alert assigned to the user

User access token was changed

Key file added

Application activated with an activation code

Reserve license key added

Reserve license key deleted

License renewed

Invalid activation code

Activation code blocked

Activation code or key file deleted

Activation code or key file replaced

EPS exceeded

License expiration and start of grace period

End of grace period

Expired activation code or key file 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

Partition deleted by absolute retention period

Partition deleted by relative retention period

Partition moved to cold storage

Error 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 was unsuccessful

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
Asset was created
Asset was deleted
Asset category was added
Asset category was deleted
Settings were updated
Tenant was created
Tenant was enabled
Tenant was disabled
Other tenant data was changed
The dictionary was successfully updated on the service or operation was unsuccessful
Dictionary entry added
Dictionary entry successfully deleted or the operation failed
Incident successfully created or operation failed
Incident successfully closed or operation failed
Incident assigned to a user or operation failed
Alert linked to an incident or alert unlinked from an incident, or operation failed
Metrics alert registered for service or operation failed
Event linked to an alert or unlinked from an alert, or operation failed
Response in Active Directory
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
Query sent to KIRA
KICS/KATA response
Kaspersky Automated Security Awareness Platform response
KEDR response
Importing MITRE ATT&CK techniques and tactics
Backing up KUMA Core and restoring from backup
Backing up partitions, restoring from backup, importing partitions from an external KUMA installation
Correlation rules
Sending test events to KUMA
Time format
Mapping fields of predefined normalizers
Deprecated resources
Generating events for testing a normalizer
Glossary
Aggregation
Cluster
Collector
Connector
Correlation rule
Dashboard
Enrichment
Event

- Filter
- KUMA web interface
- Network port
- Normalization
- Normalizer
- Parsing
- Raw event
- Report
- Role
- SELinux (Security-Enhanced Linux)
- SIEM
- STARTTLS
- UserPrincipalName

Information about third-party code

Trademark notices

The MITRE Corporation Terms of Use

Known limitations and issues

Kaspersky Unified Monitoring and Analysis Platform Help

New features

- What's new in KUMA

Hardware and software requirements

- Hardware and software requirements

Getting started

- Architecture of the solution
- Installation and removal
- Managing tenants
- Monitoring event sources

Managing the KUMA web interface

- Administrator's guide
- User guide

Additional features

- Interaction with external systems via API

Licensing

- KUMA licensing

Contacting Technical Support

- How to get technical support

About Kaspersky Unified Monitoring and Analysis Platform

Kaspersky Unified Monitoring and Analysis Platform (hereinafter KUMA or "solution") is an integrated software solution that combines the following functionality:

- Receiving, processing, and storing information security events
- Analyzing and correlating incoming data
- Searching in received events
- Creating notifications about detected indicators of information security threats

The solution is built on a microservice architecture. This means that you can create and configure only those microservices (hereinafter also "services") that you need, which lets you use KUMA as a log management system or as a full-fledged SIEM system. In addition, flexible routing of data feeds lets you use third-party services for additional event processing.

The update functionality (including anti-virus signature updates and code base updates) may not be available in the solution in the territory of the USA.

What's new

Kaspersky Unified Monitoring and Analysis Platform 4.2 introduces the following features and improvements:

- 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.
- After upgrading KUMA from version 4.0 to 4.2, Kubernetes cluster components can be upgraded.
- You can now create custom roles to supplement the predefined roles. For custom roles, you can create unique sets of rights and permissions and indicate the functional areas that the user needs.

- 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** type connector for obtaining data from arbitrary REST APIs with JSON support, filtering by ID, and iteration mode.
- Now you can run SQL requests 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).
- The table in the **Alerts** section has a new **Additional info** column, which displays the value of the correlation event field selected when configuring alert filling. For some fields, associated values are automatically displayed along with the main value in the alert card.
- 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.
- Now you can format the display of an event in the **Event details** area. By clicking the **Format** button, you can switch the event display format between single-line and multi-line (indented) XML or JSON format.
- 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.

Distribution kit

The distribution kit includes the following files:

- kuma-ansible-installer-<build number>.tar.gz, used to install KUMA without using Kubernetes. You can perform a distributed installation with the KUMA Core in a Raft-based cluster.
- kuma-ansible-installer-k8s-<build number>.tar.gz, used for installing KUMA in a high-availability configuration based on Kubernetes.
- Files containing information about the version (release notes) in Russian and English

Hardware and software requirements

Recommended hardware

This section lists the hardware requirements for processing an incoming event stream in KUMA at various Events per Second (EPS) rates.

The table below lists the hardware and software requirements for installing the KUMA components, assuming that the ClickHouse cluster only accepts INSERT queries. Hardware requirements for SELECT queries are calculated separately for the particular database usage profile of the customer.

Recommended hardware for ClickHouse cluster storage

When designing the storage cluster, the EPS, the average event size, and the storage depth are not the only things you must take into account. You must also be mindful of special considerations that are presented in this article for your convenience in the following sections:

- Disk subsystem
- Network interface
- RAM
- Usage profile

Disk subsystem

Insert queries (write access) constitute the majority of the load that KUMA places on ClickHouse:

- Events arrive in the database in an almost constant stream.
- Inserts are small and frequent by ClickHouse standards.
- ClickHouse has high write amplification because of the constant merging of parts of partitions in the background.
- A partition part under 1 GB in size is stored as a single file. When the part reaches 1 GB, wide column format is applied, wherein each column of the table is represented by two files, .dat and .mrk. In this case, overwriting a part to merge it with another part requires writing to 384 files + fsync for each, which is a lot of IOPS.

Of course, ClickHouse handles search queries at the same time, but there are much fewer of those than write operations. Considering the load profile, we recommend organizing the disk subsystem in the following way:

1. Use DAS (Direct Attached Storage) with NVME or SAS/SATA interfaces, avoiding SAN/NAS solutions.
2. If the nodes of the ClickHouse cluster are deployed in a virtual environment, the disk array must be directly mounted to the /opt/kaspersky/kuma directory. Avoid virtual disks.

3. Ideally, SSD (SATA/NVME) should be used, if possible. Any server-grade SSDs will outperform HDDs (even 15k RPM). For SSDs, use RAID-10 or RAID-0; for RAID-0, make sure that replication in ClickHouse is used.

4. If SSDs cannot be used, use 15k-RPM SAS HDDs in RAID-10.

5. A hybrid option is possible: storing hot data (for example, for the last month) on SSDs, and cold data on HDDs. This arrangement makes sure the SSD arrays always handle the write operations.

6. A software RAID array (madm) will always outperform a hardware RAID controller and offer more flexibility. When using RAID-0 or RAID-10, it is important that the stripe size be 1 MB. In hardware RAID controllers, the default stripe size is often 64 KB, and the maximum supported value is 256–512 KB. For RAID-10, near-layout is optimal for writing, and far-layout is optimal for reading. You must keep this in mind when using a hybrid SSD + HDD configuration.

7. We recommend EXT4 as the file system. If, for any reason, this file system cannot be used, you may use the XFS file system instead. We recommend enabling the 'noatime' mount option for either of these file systems.

Network interface

1. On each node of the cluster, a network interface with a bandwidth of at least 10 Gbps must be installed and appropriately switched. Considering write amplification and replication, 25 Gbps is ideal.

2. If necessary, replication processes can use a separate network interface that does not handle insert and search traffic.

RAM

ClickHouse recommends keeping the RAM-to-stored-data ratio at 1:100.
But how much RAM is needed if each node is supposed to store 50 TB, given that the event storage depth is often dictated by regulatory requirements, while search queries are not performed to the full storage depth? Thus, the recommendation can be interpreted as follows: if the average query depth involves scanning partitions with a total volume of 10 TB, then you need 100 GB of RAM. This is a general recommendation for broad analytical queries.
In the general case, more free RAM on the server improves the performance of search queries for the most recent events because the contents of the partition files will stay in operating system's page cache, so file content will be read from RAM, and not from disk.

Usage profile

The hardware requirements are designed for the database to consume a stream of a certain EPS when the database is at rest (no search queries are being executed, only inserts). At the KUMA deployment stage, it is not always possible to accurately answer the following questions:

1. What search and analytic queries will be performed in the system?

2. What is the intensity, concurrency, and depth of search queries?

3. What is the actual performance of the cluster nodes?

Thus, it is perfectly normal for a KUMA ClickHouse cluster to evolve over time. If an increase of EPS or intensity/resource consumption of search queries makes the cluster unable to cope with the load, you can add more shards (horizontal scaling) or improve the disk subsystems/CPU/RAM on the cluster nodes.

The configuration of the equipment must be chosen based on the system load profile. You can use the "All-in-one" configuration for an event stream of under 10,000 EPS and when using graphical panels supplied with the system.

KUMA supports Intel and AMD CPUs with SSE 4.2 and AVX instruction set support.

	Up to 3,000 EPS	Up to 10,000 EPS	Up to 20,000 EPS	Up to 50,000 EPS
KUMA Core requirements	–	–	One device. Device characteristics: • At least 12 threads or vCPUs. • At least 12 GB of RAM. • At least 500 GB in the /opt directory. • Data storage type: SSD. • Data transfer rate: at least 53 Mbps.	One device. Device characteristics: • At least 16 threads or vCPUs. • At least 24 GB of RAM. • At least 1 TB in the /opt directory. • Data storage type: SSD. • Data transfer rate: at least 134 Mbps.
Collector requirements	–	–	One device. Device characteristics: • At least 16 threads or vCPUs. • At least 24 GB of RAM. • At least 500 GB in the /opt directory. • Data storage type: SSD. • Data transfer rate: at least 53 Mbps.	Two devices. Characteristics of each device: • At least 16 threads or vCPUs. • At least 24 GB of RAM. • At least 500 GB in the /opt directory. • Data storage type: SSD. • Data transfer rate: at least 134 Mbps.
Correlator requirements	–	–	One device. Device characteristics: • At least 16 threads or vCPUs. • At least 32 GB of RAM. • At least 500 GB in the /opt directory. • Data storage type: SSD. • Data transfer rate: at least 53 Mbps.	One device. Device characteristics: • At least 16 threads or vCPUs. • At least 32 GB of RAM. • At least 500 GB in the /opt directory. • Data storage type: SSD. • Data transfer rate: at least 134 Mbps.
Event router requirements	–	CPU: 3 cores RAM: 240 MB.	CPU: 6 cores RAM: 250 MB.	CPU: 9 cores RAM: 7000 MB.
Keeper requirements	–	–	Three devices. Characteristics of each device: • At least 6 threads or vCPUs. • At least 12 GB of RAM. • At least 100 GB in the /opt directory. • Data storage type: SSD. • Data transfer rate: at least 53 Mbps.	Three devices. Characteristics of each device: • At least 6 threads or vCPUs. • At least 12 GB of RAM. • At least 100 GB in the /opt directory. • Data storage type: SSD. • Data transfer rate: at least 134 Mbps.

	Up to 3,000 EPS	Up to 10,000 EPS	Up to 20,000 EPS	Up to 50,000 EPS
Storage requirements	–	–	Two devices. Characteristics of each device: • At least 24 threads or vCPUs. • At least 64 GB of RAM. • At least 16.5 TB in the /opt directory. • Data storage type: SSD*. The recommended transfer rate between ClickHouse nodes is at least 10 Gbps if the data stream is equal to or exceeds 20,000 EPS.	Four devices. Characteristics of each device: • At least 24 threads or vCPUs. • At least 64 GB of RAM. • At least 41.3 TB in the /opt directory. • Data storage type: SSD*. The recommended transfer rate between ClickHouse nodes is at least 10 Gbps if the data stream is equal to or exceeds 20,000 EPS.
Operating systems	• Ubuntu 22.04 LTS • Ubuntu 24.04 LTS • Oracle Linux 8.x • Oracle Linux 9.x • Astra Linux Special Edition RUSB.10015-01 1.7.x (starting from version 2021-1126SE17 update 1.7.1). • Astra Linux Special Edition RUSB.10015-01 1.8.x (starting with version 2025-0114SE18MD, urgent update 1.8.1.UU.2) • RED OS 7.3.4 • RED OS 8 • MosOS 15.5			
TLS ciphersuites	TLS versions 1.2 and 1.3 are supported. Integration with a server that does not support the TLS versions and ciphersuites that KUMA requires is impossible. Supported TLS 1.2 ciphersuites: • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 Supported TLS 1.3 ciphersuites: • TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256			

Recommended configurations for different load levels

The table below describes the configurations and recommended resources for KUMA components depending on the EPS level, including storage requirements and sizing recommendations.

Recommended hardware configurations and resource consumption by KUMA depending on EPS

Setting	Up to 3,000 EPS	Up to 10,000 EPS	Up to 20,000 EPS	Up to 50,000 EPS
Installation type	Installation on a single server	Installation on a single server	Distributed installation on multiple servers	Distributed installation on multiple servers
Number of servers and their purpose	One server that hosts all system components.	One server that hosts all system components.	One server for the KUMA Core. One server for the collector One server for the correlator. Three dedicated servers with the keeper role. Two servers for the Storage.	One server for the KUMA Core. Two servers for the Collector. One server for the correlator. Three dedicated servers with the keeper role. Four servers for the storage.

Setting	Up to 3,000 EPS	Up to 10,000 EPS	Up to 20,000 EPS	Up to 50,000 EPS
Server hardware	At least 16 threads or 16 vCPUs. At least 32 GB of RAM. At least 500 GB in the /opt directory. Data storage type: SSD. Data transfer rate: at least 100 Mbps.	At least 24 threads or 24 vCPUs. At least 64 GB of RAM. At least 500 GB in the /opt directory. Data storage type: SSD. Data transfer rate: at least 100 Mbps.	See the specifications for KUMA components in the table above. The storage requires fast SSDs, high IOPS, at least 1 Gbps of network bandwidth.	Same specifications as for EPS 20,000, but with twice the number of collectors and storage servers. Higher requirements for network connectivity between all nodes: at least 1 Gbps of network bandwidth is recommended.
Storage configuration (ClickHouse)	One shard, one replica. Disk buffer enabled.	One shard, one replica. Disk buffer enabled.	Two shards, two replicas in each. If high availability is not required, one server with one replica per shard may be used.	Two shards, two replicas in each. If high availability is not required, two servers with one replica per shard may be used.

Additional recommendations for specifying and sizing the configuration:

- For the KUMA Core component, at over 50,000 assets, we recommend provisioning more resources: for every additional 50,000 assets, add 2 threads (or 2 vCPUs) and 4 GB of RAM. If the system manages more than 100 services, you must consider the additional load: for every 100 services, we recommend also increasing the computational resources of the KUMA Core by 2 threads (or 2 vCPUs).
- If users actively engage with the system, including frequent generation of reports, using dashboards, and running custom SQL queries, you need to adjust the sizing of both the KUMA Core and the storage because such activity substantially increases the load on both components.
- When choosing the disk subsystem for the storage (ClickHouse), only SSD drives may be used. An exception is possible if you do not plan to run aggregation SQL queries with a depth of more than 24 hours. In this case, you may use HDDs with the following minimum specifications: 15,000 RPM, SAS interface, RAID-10.
- When placing components in HDFS, you may use HDD regardless of the queries that you run.
- Events are exported via temporary files that are saved in the /opt/kaspersky/kuma/core/tmp/ directory. These files are automatically deleted after 10 days. If you plan to export a large amount of events, you must allocate additional space.

Working in virtual environments

The following virtual environments are supported for installing KUMA:

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

Working in cloud environments

KUMA can work in a cloud infrastructure. The system can be installed on virtual machines following the IaaS (infrastructure-as-a-service) model.

For a cloud infrastructure, we recommend using the single-server configuration for 3000 EPS and 10,000 EPS. Virtual machines must satisfy the hardware and software requirements of a regular installation.

When choosing the disk subsystem of the server, use the "number of input/output operations (IOPS)" parameter as the reference. The recommended minimum value is 1000 IOPS.

Recommendations for the Collector service

Consider that for event processing efficiency, the CPU core count is more important than the clock rate. For example, eight CPU cores with a medium clock rate can process events more efficiently than four CPU cores with a high clock rate.

Consider also that 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). The utilization of computation resources by KUMA depends on the type of events being parsed and the efficiency of the normalizer.

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.

Recommendations for storage servers

You must use high-speed protocols, such as Fibre Channel or iSCSI 10G for the connection of the data storage system to storage servers. We do not recommend using application-level protocols such as NFS or SMB to connect data storage systems.

On ClickHouse cluster servers, we recommend using the ext4 file system.

If you are using RAID arrays, we recommend using RAID 0 for high performance, or RAID 10 for high performance and high availability.

To ensure high availability and performance of the data storage subsystem, we recommend making sure that all ClickHouse nodes are deployed strictly on different disk arrays.

If you are using a virtualized infrastructure to host system components, we recommend deploying ClickHouse cluster nodes on different hypervisors. You must prevent any two virtual machines with ClickHouse from running on the same hypervisor.

For high-load KUMA installations, we recommend installing ClickHouse on physical servers.

Recommendations for the Metrics service

The Metrics service monitors the performance of KUMA (the **Metrics** section) and monitoring of event streams from event sources (the **Source status** section).

Many factors affect the computing power requirements of the Metrics service, including the following:

- The total number of created KUMA services
- The settings of the event source monitoring service

The system workload profile must be considered when choosing the hardware configuration of the Metrics service.

Recommended hardware for the Metrica service

Sizing the solution depending on the number of services

The table below provides server hardware requirements for the Metrics service, depending on the number of services in the KUMA instance. The figures were calculated for an event stream of 20,000 EPS. The number of CPUs and the amount of RAM in the table are rounded up to the nearest integer number.

The results in the table were obtained with the following parameters:

- Source discovery settings: default.
- Monitoring policy settings:
  - **Type**: byCount
  - **Lower limit**: 10
  - **Upper limit**: 1000
  - **Count interval**: 10 minutes
  - **Evaluation interval**: 5 minutes

Sizing the solution depending on the number of services

Number of services	Number of CPUs (recommended)	RAM (GB) (recommended)
50	1	3
100	1	3
500	1	4

According to the table, the increase in the number of services in the KUMA instance only slightly increases the hardware requirements.

Sizing the solution depending on the event source status

The hardware requirements (CPUs, RAM, and disk space) are affected by the following parameters displayed or set in the **Metrics** and **Source status** sections:

- Settings of event source detection

  The hardware requirements are affected by the settings specified for event source detection: the number of fields and their properties. If you increase the number of fields or specify fields that contain a large number of unique values, this may result in KUMA detecting a large number of event sources. Therefore, increasing the number of event sources increases the hardware requirements.

  For example, the set of fields "DeviceProduct, DeviceHostName, DeviceAddress" detects the firewall "Firewall, fw-1.example.com, 192.0.2.1" as an event source. However, if you specify the set of fields "DeviceProduct, DeviceHostName, DeviceAddress, Name", KUMA detects a much larger number of event sources:

  - Firewall, fw-1.example.com, 192.0.2.1, DNS
  - Firewall, fw-1.example.com, 192.0.2.1, IPS
  - Firewall, fw-1.example.com, 192.0.2.1, TRAFFIC
  - Firewall, fw-1.example.com, 192.0.2.1, TREAT

  Consider that a noticeable impact on hardware requirements begins at 100,000 event sources.

- The number of event sources to which monitoring policies have been assigned, as well as the data counting period and the evaluation interval

  KUMA periodically receives and records information about the data feed from each event source to which a monitoring policy has been assigned with a frequency specified in the policy.

- Number of monitoring policies assigned to each event source

- Average data point size (value of the **Bytes per point** metric)

  Using the value of the **Bytes per point** (bpp) metric of the Metrics service, you can calculate how much disk space you need to store the metrics. To do this, you can use the following formula:

  hdd_sizing = ROUNDUP (bpp * (retention_sec/interval_sec) * EPS * 1,1),

  where:

  - hdd_sizing is the required amount of disk space
  - ROUNDUP is the rounding up function
  - bpp is the value of the Bytes per point metric
  - retention_sec is the metrics retention period in seconds
  - interval_sec is the information retrieval interval, equal to 15 seconds

- EPS is the average EPS across sources, approximately equal to the value of the **Processing EPS** metric on the collector
- 1.1 is a factor of safety that adds 10% to the calculated disk space.

- The stream of events processed by all collectors in KUMA

  The following table provides the hardware requirements for sizing the solution, depending on the number of event sources and monitoring policies in a KUMA instance.

  According to the table, the hardware requirements substantially increase when the number of event sources is large and a large number of monitoring policies is assigned to event sources. This data is listed in the bottom of the table.

  The results in the table were obtained with the following parameters:

  - Source discovery settings: default.
  - Monitoring policy settings:
    - **Type**: byCount
    - **Lower limit**: 10
    - **Upper limit**: 1000
    - **Count interval**: 10 minutes
    - **Evaluation interval**: 5 minutes

Sizing the solution depending on the number of event sources and monitoring policies

Number of event sources	Number of event sources monitored by monitoring policies	EPS	Number of monitoring policies per source	Number of CPUs (recommended)	RAM (GB) (recommended)	Disk space (GB) for 30 days of metrics retention
10,000	0	25,000	0	1	2	13
25,000	0	25,000	0	1	2	13
50,000	0	25,000	0	1	3	13
100,000	0	25,000	0	1	3	14
250,000	0	25,000	0	1	3	13
500,000	0	25,000	0	1	5	13
1,000,000	0	25,000	0	1	7	11
10,000	2000	5000	1	1	3	4
10,000	5000	5000	1	1	3	5
10,000	2000	10,000	1	1	3	8
10,000	2000	10,000	2	1	3	10
10,000	5000	10,000	1	1	3	9
10,000	5000	25,000	1	1	3	24
10,000	10,000	5000	1	1	4	5
10,000	10,000	5000	2	1	5	5
50,000	10,000	5000	1	1	4	4
50,000	10,000	10,000	1	1	5	7
50,000	25,000	25,000	1	1	6	20

100,000	20,000	5000	1	1	6	4
100,000	20,000	10,000	1	1	6	8
100,000	100,000	5000	1	4	19	5
250,000	50,000	10,000	1	3	12	7
100,000	100,000	5000	2	6	29	5
250,000	250,000	5000	1	10	40	4
250,000	250,000	5000	2	17	68	5

Requirements for agent devices

You must install agents on network infrastructure devices that will send data to the KUMA collector. Device requirements are listed in the following table.

	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 2012 support is limited. • Microsoft Windows Server 2012 R2 Microsoft Windows 2012 R2 support is limited. • Microsoft Windows Server 2016 • Microsoft Windows Server 2019 • Microsoft Windows Server 2022 • Microsoft Windows Server 2025 • Microsoft Windows 10 • Microsoft Windows 11	• Oracle® Linux 8.x • Oracle Linux 9.x • Alma Linux 9.5 • Astra Linux Special Edition RUSB.10015-01 1.7.x (starting with version 2021-1126SE17, update 1.7.1) with glibc 2.28 (Debian GLIBC 2.28-10+deb10u4+ci202407161455+astra12). • Astra Linux Special Edition RUSB.10015-01 1.8.x (starting with version 2025-0114SE18MD, update 1.8.1.UU.2) with glibc 2.36 (Debian GLIBC 2.36-9+deb12u7+ci202405171200+astra5). • CentOS Linux release 8.2.2004 (Core) with glibc 2.28 (GNU libc). • CentOS Linux release 7.7.1908 (Core) with glibc 2.17 (GNU libc). • Debian 12 c with glibc 2.36 (Debian GLIBC 2.36-9+deb12u7). • Red Hat Enterprise Linux 9.5 (Plow) with glibc 2.34 (GNU libc). • Red OS 8. • Rocky Linux 9.3 (Blue Onyx) with glibc 2.34 (GNU libc) • Rocky Linux 9.5. • SuSE 15.6. • Ubuntu 24.04.2 LTS with glibc 2.39 (Ubuntu GLIBC 2.39-0ubuntu8.4) • Ubuntu 22.04.4 LTS with glibc 2.35 (Ubuntu GLIBC 2.35-0ubuntu3.8) • MosOS 15.5

Hardware requirements for installing KUMA agents with WEC, ETW, and WMI connectors

KUMA agents installed on Windows devices can use various types of connectors to receive events: WEC (Windows Event Collector), ETW (Event Tracing for Windows), and WMI (Windows Management Instrumentation). The performance of the agents and their impact on the system depend on the limitations imposed by Windows and on the amount and type of events processed (see the table below).

Requirements for agents by connector type

Connector type	Maximum performance (EPS)	Limitations imposed by Windows	Limitations imposed by KUMA	Recommended agent configuration	CPU and RAM usage
WEC	Up to 2500 events per second	When writing to system logs (Application, System, Hardware Events), Windows limits the rate to approximately 2500 events per second. When writing to multiple logs at the same time, performance is further reduced: up to 1100 EPS with 2 logs, up to 800 EPS with 3 logs, and up to 500 EPS with 4 logs.	The agent can process events in the range of 1500–2500 EPS without lag.	CPU: 1 core with 2 threads or 2 vCPUs. RAM: 4 GB.	Average CPU load by the agent: up to 0.18 (equivalent to approximately 0.28% of the total system performance). RAM usage: up to 209 MB.
ETW	Up to 700 events per second	Only one system log is supported: **Microsoft → Windows → DNS-Server**. The Windows operating system does not allow writing more than 700 events per second to this log.	No limitations on the KUMA side have been identified, however, when EPS exceeds 700, the agent does not receive data due to Windows restrictions.	CPU: 1 core with 2 threads or 2 vCPUs. RAM: 4 GB.	Average CPU load by the agent: up to 0.267 (equivalent to approximately 0.42% of the total system performance). RAM usage: up to 186 MB.
WMI	Up to 5000 events per second	You can circumvent Windows restrictions by connecting multiple devices. Each additional device increases the total event volume that can be processed by 1000–1500 EPS.	Upon reaching 5000 EPS, the agent starts processing events with a lag.	CPU: 1 core with 2 threads or 2 vCPUs. RAM: 4 GB.	Average CPU load by the agent: up to 0.61 (equivalent to approximately 0.91% of the total system performance). RAM usage: up to 311 MB.

Hardware requirements for KUMA agents on Linux devices depending on EPS

KUMA agents installed on Linux devices can use various types of transports. The hardware requirements for KUMA agents depend on the type of transport and the EPS (see the table below).

Requirements for KUMA agents on Linux devices depending on EPS

Transport type	10,000 EPS	20,000 EPS	60,000 EPS	100,000 EPS
HTTP	CPU: 3 cores RAM: 160 MB.	CPU: 5 cores RAM: 160 MB.	CPU: 6 cores RAM: 260 MB.	CPU: 6 cores RAM: 300 MB.

Hardware requirements for using the Score AI service and asset status

CPU: 2 cores, 2.7 GHz.

RAM: 8 GB

Storage: SSD or HDD.

Free disk space: 10 GB for the service itself and space for the OS.

Requirements for client devices for managing the KUMA web interface

CPU:

- Intel® Core™ i3, 8th generation, 32-bit
- AMD Ryzen 3, 32-bit

RAM: 8 GB

Supported browsers:

- Google™ Chrome™ 110 or later
- Mozilla™ Firefox™ 115 or later

Device requirements for installing KUMA on Kubernetes

Minimum configuration of a Kubernetes cluster for deployment of a high-availability KUMA configuration:

- 1 load balancer node (not part of the cluster)
- 3 controller nodes
- 2 worker nodes

The minimum hardware requirements for devices for installing KUMA on Kubernetes are listed in the table below.

Minimum hardware requirements for installing KUMA on Kubernetes

	Balancer	Controller	Worker node
CPU	1 core with 2 threads or 2 vCPUs.	1 core with 2 threads or 2 vCPUs.	12 threads or 12 vCPUs.
RAM	At least 2 GB	At least 2 GB	At least 24 GB
Free disk space	At least 30 GB	At least 30 GB	For new instances of KUMA: at least 600 GB of free disk space in the /opt directory. For existing KUMA instances: if 1 TB or more disk space has been previously allocated in the /opt directory, we do not recommend reducing disk space without expert advice from Technical Support. At least 32 GB in the /var/lib directory.
Network bandwidth	10 Gbps	10 Gbps	10 Gbps

KUMA interface

The solution is managed using the web interface.

The window of the solution web interface contains the following:

- Sections in the left part of the solution web interface window
- Tabs in the upper part of the solution web interface window for some sections of the solution
- The workspace in the lower part of the solution web interface window

The workspace displays the information that you choose to view in the sections and on the tabs of the solution web interface window. It also contains controls that you can use to configure the display of the information.

While managing the solution web interface, you can use shortcut keys to perform the following actions:

- In all sections: close the window that opens in the right side pane—**Esc**.
- In the **Events** section:
  - Switch between events in the right side pane—↑ and ↓.
  - Start a search (when focused on the query field)—**Ctrl/Command**+**Enter**.
  - Save a search query—**Ctrl/Command**+**S**.

Compatibility with other applications

Kaspersky Endpoint Security for Linux

If KUMA components and the Kaspersky Endpoint Security for Linux application are installed on the same server, the report.db directory may grow very large and even take up the entire drive space. In addition, Kaspersky Endpoint Security for Linux scans all KUMA files by default, including service files, which may affect performance. To avoid these problems:

- Upgrade Kaspersky Endpoint Security for Linux to version 12.0 or later.
- We do not recommend enabling the network components of Kaspersky Endpoint Security for Linux.
- Add the following directories to general exclusions and to on-demand scan exclusions:

  a. On the KUMA Core server:

  - /opt/kaspersky/kuma/metrics/ — directory with data of the Metrics service.

  b. On the storage server:

  - /opt/kaspersky/kuma/clickhouse/—the ClickHouse directory.
  - /opt/kaspersky/kuma/storage/<storage ID>/buffers/—directory with storage buffers.

c. On the correlator server:

- /opt/kaspersky/kuma/correlator/<correlator ID>/data/—directories with dictionaries.
- /opt/kaspersky/kuma/correlator/<correlator ID>/lists—directories with active lists.
- /opt/kaspersky/kuma/correlator/<correlator ID>/ctxtables—directories with context tables.
- /opt/kaspersky/kuma/correlator/<correlator ID>/buffers—directory with buffers.

d. On the collector server:

- /opt/kaspersky/kuma/collector/<collector ID>/buffers—directory with buffers.
- /opt/kaspersky/kuma/collector/<collector>/data/—directory with dictionaries.

e. Directories with logs for each service.

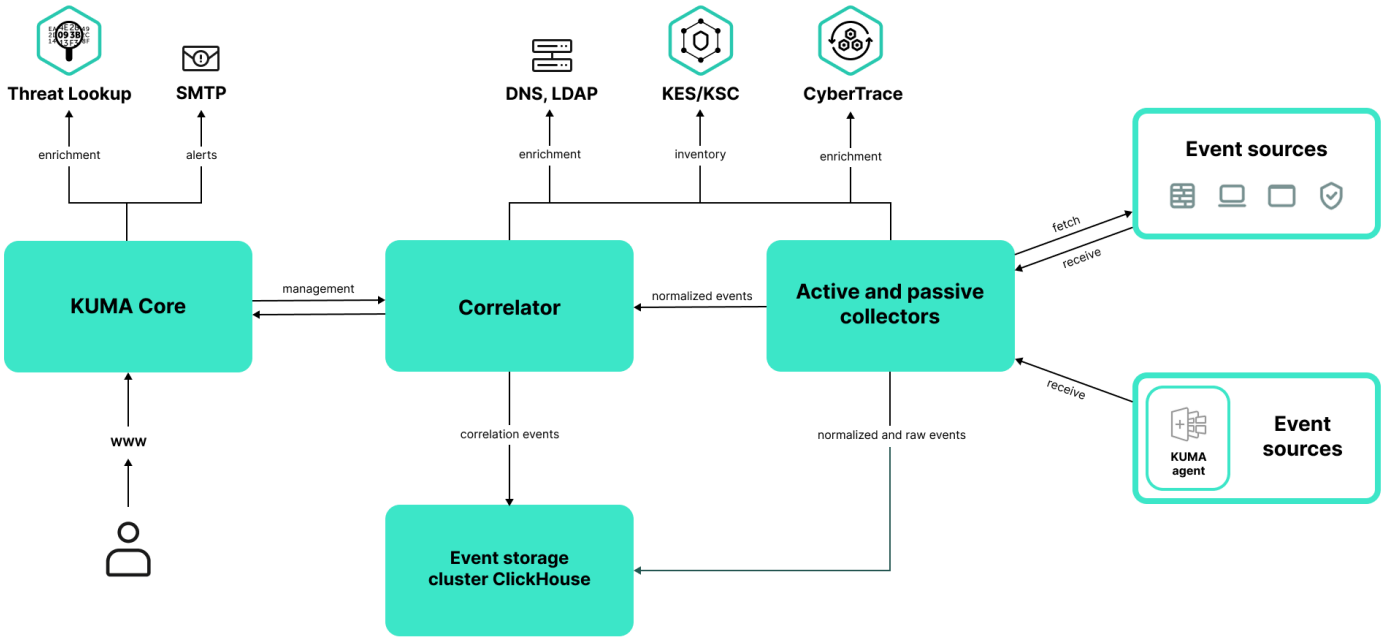
For more details on scan exclusions, please refer to the Kaspersky Endpoint Security for Linux Online Help.

Architecture of the solution

The standard installation of the solution includes the following components:

- The *Core* that includes a graphical user interface for monitoring and managing the settings of system components.
- *Agents* that are used to forward raw events from servers and workstations to KUMA destinations.
- One or more *Collectors* that receive messages from event sources and parse, normalize, and, if necessary, filter and/or aggregate them.
- *Event routers* that receive events from collectors and apply the configured filters to route the events to the configured destinations. In this way, these services balance the load on the network links.
- The *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 holds normalized events and registered alerts.

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



KUMA architecture

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 provides a graphical user interface that is intended for everyday use as well as for configuring the system as a whole.

The Core allows you to:

- Create and configure services (or components) of the solution, as well as integrate the necessary software into the system.
- Manage services and user accounts of the solution in a centralized way.
- Visualize solution performance statistics.
- Investigate security threats based on the received events.

Collector

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

For each collector, one connector and one normalizer must be configured. You can also configure any number of additional normalizers, filters, enrichment rules, and aggregation rules. For the collector to send normalized events to other services, you must add destinations. Normally, two destinations are used: a storage and a correlator.

The collector iterates over the following steps:

1. **Receive messages from event sources**

   To receive messages, you must configure an active or passive connector. A passive connector only listens for messages from an event source, while an active connector actively polls an event source, such as a database management system.

   Connectors can have different types. The choice of connector type depends on the transport protocol used for messaging. For example, if your event source sends messages over TCP, you must install a connector of the TCP type.

   The solution provides the following types of connectors:

   - tcp
   - udp
   - netflow
   - sflow
   - nats-jetstream
   - kafka
   - kata/edr
   - http
   - sql
   - file
   - 1c-xml

- 1c-log
- diode
- ftp
- nfs
- vmware
- wmi
- wec
- snmp-trap
- elastic
- etw

2. **Parse and normalize events**

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

   The following normalizers are available in the solution:

   - JSON
   - CEF
   - Regexp
   - Syslog (as per RFC3164 and RFC5424)
   - CSV
   - Key-value
   - XML
   - NetFlow (the same normalizer for NetFlow v5, NetFlow v9 and IPFIX)
   - NetFlow v5;
   - NetFlow v9;
   - SQL
   - IPFIX (v10).

3. **Filter normalized events**

   You can configure filters to identify events that satisfy certain conditions and only send such events for processing.

4. **Enrich and convert normalized events**

Enrichment rules let you add supplementary information from internal and external sources to the events. The solution can use the following enrichment sources:

- 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 solution offers the following conversion methods:

- lower: convert all characters to lower case.
- upper: convert all characters to upper case.
- regexp: extract a substring using RE2 regular expressions.
- substring—gets a substring based on the specified numbers of the start and end positions.
- replace: replace some text with a string.
- trim: delete the specified characters.
- append: add characters to the end of the field value.
- prepend: adds characters to the beginning of the field value.

5 **Aggregate normalized events**

You can configure aggregation rules to avoid sending many events of the same kind to the storage and/or 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 **Transferring normalized events**

Having passed through all processing steps, the event is sent to the configured destinations.

Correlator

A *correlator* is a component of the solution that analyzes normalized events. As part of the correlation process, an event can be correlated with information from active lists and/or dictionaries.

The correlation analysis produces information that can be used for the following purposes:

- Generating alerts.
- Notifying about alerts.
- Managing active list content.
- Sending correlation events to configured destinations.

Events are correlated in real time. The operating principle of the correlator is based on signature analysis of events. This means that every event is processed in accordance with the correlation rules set by the user. When the application detects a sequence of events that match the correlation rule, the solution creates a correlation event and sends it to the Storage. The correlation event can also be sent to a correlator to be analyzed again, which lets you configure correlation rules that trigger on prior analysis results. Products 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 balancing the load on services. In this arrangement, collectors will send normalized events to all available correlators.

The correlator operation algorithm has the following steps:

1. **Get an event**

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

2. **Apply correlation rules**

   You can configure correlation rules to trigger on a single event or a sequence of events. If correlation rules do not detect an alert, the event processing ends here.

3. **Respond to an alert**

   You can configure what the solution does when an alert is detected. The following actions are available in the solution:

   - Event enrichment
   - Operations with active lists
   - Sending notifications
   - Saving a correlation event

4. **Send a correlation event**

   When the application detects a sequence of events that match the correlation rule, the solution creates a correlation event and sends it to the storage. At this point, the correlator is done processing the event.

Storage

A KUMA *storage* is used to store normalized events and ensure that KUMA can quickly and reliably access these events to extract analytical data. Access speed and high availability are made possible by 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*. Spaces let you structure the data in the cluster and, for example, store events of a certain type together.

Basic entities

This section introduces the entities that KUMA works with.

Tenant

KUMA has a multitenancy mode in which one instance of the KUMA solution installed in the infrastructure of the main organization (main tenant) enables isolation of branches (tenants) so that they receive and process their own events.

The system is managed centrally through the shared web interface, however, the tenants operate independently of each other and have access only to their own resources, services, and settings. Events of tenants are stored separately.

A user can have access to multiple tenants at the same time. You can also select which tenants' data you want to be displayed in sections of the KUMA web interface.

Two tenants are created by default in KUMA:

- The Main tenant contains resources and services that belong to the main tenant. Only the general administrator has access to these resources.

- The Shared tenant is where the general administrator can place resources, asset categories, and monitoring policies that are available to users of all tenants. Access to the Shared tenant can be **restricted** for individual users.

  If in the settings of a user, the **Hide shared resources** check box is selected, that user cannot gain access to the Shared folder belonging to the Shared tenant in the KUMA web interface in the **Resources** → **<resource type>** section. This means that the user cannot view, edit, or otherwise use shared resources. The user is also unable to export shared resources and resource sets that incorporate resources from the Shared tenant, either through the web interface or through the REST API.

  If any of the services available to the user use shared resources, the names of these resources are displayed in the service settings, but user cannot view or modify the resources. The content of active lists is available to the user even if the resource of this active list is shared.

The limitation does not apply to shared asset categories. Shared resources are also always available to users with the general administrator role.

Event

Events are information security events registered on the monitored elements of the corporate IT infrastructure. For example, events include login attempts, interactions with a database, and information sent by sensors. Each individual event may appear meaningless, but taken together, they paint a bigger picture of network activity that can help you identify security threats. This is the core functionality of KUMA.

KUMA receives events from logs and restructures them by bringing data from heterogeneous sources to a uniform format (this process is called normalization). The events are then filtered, aggregated, and sent to the correlator service for analysis and to the storage service where they are retained. When KUMA recognizes a specific event or a sequence of events, it creates *correlation events*, which 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. 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 occurrence can be combined into an incident.

For convenience of investigating alerts and processing incidents, make sure that time is synchronized on all devices involved in the event life cycle (event sources, KUMA servers, client hosts) with the help of Network Time Protocol (NTP) servers.

Throughout their life cycle, events undergo conversions and may be named differently. The following is an outline of the life cycle of a typical event:

The first steps are carried out in a collector.

1. Raw event. The original message from an event source received at a KUMA connector is called a *raw event*. This message is unprocessed, and KUMA cannot use it yet. To make it usable, it must be normalized to fit the KUMA data model. This happens at the next stage.

2. Normalized event. A normalizer transforms the data of the raw event to make it fit the KUMA data model. After this transformation, the original message turns into a *normalized event*, which KUMA can analyze. From this point on, KUMA handles only normalized events. Raw events are no longer used, but they can be kept as a part of normalized events inside the `Raw` field.

   The solution 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. Destination. After the collector has processed the event, the event can be sent to other KUMA services: a correlator and/or storage.

Subsequent steps of the event life cycle take place in the correlator.

The following event types are distinguished:

1. Base event. An event that has been 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. These act as base events and are processed in the same way, but In addition to all of the parameters of the parent events (the events that have been "merged"), an aggregated event has a counter that tells how many 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 solution 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 performed 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.

Alert

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, a warning about possible malicious activity is displayed. Security officers responsible for data protection must investigate these alerts and respond if necessary.

KUMA automatically assigns a severity level to each alert. This parameter shows how important or numerous are the processes 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.

Related events are linked to the alerts, which allows enriching alerts with data from these events. KUMA also offers drill down functionality for alert investigation.

You can use alerts to create incidents.

Alert management in KUMA is described in this section.

Incident

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 occurrence can be combined into an *incident*. This allows security officers to analyze threat manifestations in a comprehensive manner and facilitates response.

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

Incidents can be exported to NCIRCC.

Assets

Assets are network devices registered in KUMA. Assets generate network traffic when they send and receive data. KUMA can be configured to track this activity and create base events with a clear indication of where the traffic is coming from and where it is going. The event can contain source and destination IP addresses, as well as DNS names. If you register an asset with certain parameters (for example, a specific IP address), this asset is linked to all events that mention these parameters (IP address in this example).

Assets can be logically grouped. This helps keep your network structure transparent and gives you additional ways to work with correlation rules. When an event linked to an asset is processed, the category of this asset is also taken into consideration. For example, if you assign a high severity value to a certain asset category, base events involving these assets will lead to correlation events with higher severity. This in turn cascades into higher-severity alerts and, therefore, more urgency when responding to such an alert.

We recommend registering network assets in KUMA because using assets allows formulating clear and versatile correlation rules, which makes event analysis more efficient.

Asset management in KUMA is described in this section.

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 in accordance with certain rules. These modular components are assembled into resource sets for services, which in turn are used to create KUMA services.

KUMA services

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

- One part of the service is created in the KUMA web interface based on a resource set for services.
- The other part of the service is installed in the network infrastructure where KUMA is deployed, as one of KUMA 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.

The two parts of a service are linked to each other by the service ID.

KUMA agents

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

The following types of agents are provided:

- wmi agents are used to receive data from remote Windows devices using Windows Management Instrumentation. These are installed on Windows devices.
- wec agents are used to receive Windows logs from a local device using Windows Event Collector. They are installed to Windows assets.
- tcp agents receive data over TCP. These are installed on Linux and Windows devices.
- udp agents receive data over UDP. They are installed to Linux and Windows assets.
- nats-jetstream—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 HTTP. They are installed to Linux and Windows assets.
- file agents get data from a file. They are installed to Linux and Windows assets.
- ftp agents receive data over the File Transfer Protocol. They are installed to Linux and Windows assets.
- nfs agents receive data over the Network File System protocol. They are installed to Linux and Windows assets.
- snmp agents receive data using 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 receive Event Tracing for Windows data. They are installed to Windows assets.

Severity

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

The correlator automatically assigns a severity value to correlation events and alerts based on correlation rule settings. The severity of an alert also depends on the assets linked to the events being processed because correlation rules take into account the severity of these assets' 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.

If the event is linked to assets, the system uses the highest-severity asset category to calculate the severity of the correlation event or alert.

Example of formula for calculating the severity:

- If no assets with a severity level are linked to a correlation event or alert, or no assets are linked at all:
  - **Correlation event severity** = **Correlation rule severity**.
  - **Alert severity** = **Correlation rule severity**.
- If multiple assets are linked to an event or alert, then the highest severity among the categories of assets linked to the event or asset is taken as the **highest asset category severity**. If this value is greater than the severity of the correlation rule:
  - **Correlation event severity** = (**Correlation rule severity** + **Max asset category severity** + 1) / 2.
  - **Alert severity** = (**Correlation rule severity** + **Max asset category severity** + 1) / 2.

The severity of the correlation event can never be lower than the severity of the correlation rule. The correlator and collector must belong to the same tenant as the assets. Severities of the assets from other tenants are not taken into account.

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

Possible severity values:

- 1 – Low
- 2 – Medium
- 3 – High
- 4 – Critical

Licensing

This section covers the main aspects of solution licensing.

About the End User License Agreement

The *End User License agreement* is a legal agreement between you and AO Kaspersky Lab that specifies the conditions under which you can use the solution.

Read the terms of the End User License Agreement carefully before using the solution for the first time.

You can familiarize yourself with the terms of the End User License Agreement in the following ways:

- Go to the directory with the extracted installer and read the `./roles/kuma/files/LICENSE` text file.
- Go to the directory with the extracted installer and run the following command to display the text of the End User License Agreement:

  ```
  ./roles/kuma/files/kuma license --show
  ```

- On a host with any KUMA component installed (such as KUMA Core, collector, correlator, storage), run the following command to display the text of the End User License Agreement:

  ```
  /opt/kaspersky/kuma/kuma license --show
  ```

- On devices included in the kuma_storage, kuma_collector, kuma_correlator, or kuma_core groups in the inventory file, open the LICENSE file located in the `/opt/kaspersky/kuma` directory.

  On a host in the kuma_core group, you can view the End User License Agreement only if a non-cluster installation is selected.

- On the Windows agent, run the following command to display the text of the End User License Agreement:

  ```
  .\kuma.exe license --show
  ```

- On the Linux agent, go to the directory with the 'agent' executable file and run the following command to display the text of the End User License Agreement:

  ```
  ./agent license --show
  ```

You accept the terms of the End User License Agreement by confirming your acceptance of the End User License Agreement during the solution installation. If you do not accept the terms of the End User License Agreement, you must cease the installation of the solution and must not use the solution.

About the license

A *License* is a time-limited right to use the solution, granted under the terms of the End User License Agreement.

A license entitles you to the following kinds of services:

- Use of the solution in accordance with the terms of the End User License Agreement
- Getting technical support

The scope of services and the duration of usage depend on the type of license under which the solution was activated.

A license is provided when the application is purchased. KUMA behavior in case no license is available:

- After the license expires, KUMA keeps working, but with limited functionality: collectors continue to receive and process events for 7 days, after which they restart and stop receiving new events. Old events remain available. Creating and editing resources, creating and running services also becomes impossible.
- If the license key is removed, KUMA collectors stop receiving and processing new events immediately. Old events remain available. Creating and editing resources, creating and running services also becomes impossible.

To continue using KUMA with its full functionality, you need to renew your license.

We recommend that you renew your license no later than its expiration date to ensure maximum protection against cyberthreats.

About the License Certificate

A *License Certificate* Is a document that is provided to you along with a key file or activation code.

The License Certificate contains the following information about the license being granted:

- License ID or order number
- Information about the user who is granted the license
- Information about the solution that can be activated under the provided license
- Restriction on the number of licensing units (for example, the number of events that can be processed per second)
- Start date of the license term
- License expiration date or license term
- License type

About the license key

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

To activate the solution, you can add a license key by using an *activation code* or *key file* provided to you after purchasing the license. The license key is displayed in the solution interface as a unique alphanumeric sequence after you add it to the solution.

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 solution.

A license key may be active or reserve.

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

A *reserve license key* is the license key that entitles the user to use the solution, 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 key file

The *key file* is a file named license.key provided to you by Kaspersky. The key file is used to add a license key that activates the solution.

You receive a key file at the email address that you provided after purchasing KUMA.

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

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

To restore the key file, you need to do one of the following:

- Contact the license seller.
- Get a key file on the Kaspersky website based on the available activation code.

About the activation code

An *activation code* is a unique sequence of twenty Latin letters and numerals that allows you to activate the solution. Kaspersky sends you an activation code to the email address that you provided after purchasing KUMA.

To activate the solution using the activation code, the KUMA Core server needs internet access to connect to the Kaspersky activation server at https://activation-v2.kaspersky.com:443.

In the case of a closed infrastructure, you can specify a proxy server.

If the activation code has been lost, contact the Kaspersky partner from which you purchased the license.

When the activation code is deleted from KUMA, the KUMA collectors immediately stop receiving and processing new events. Old events remain available. Creating and editing resources, creating and running services also becomes impossible.

If you have activated the solution using a key file and are adding a new activation code, activation with the activation code is performed seamlessly if the parameters of the activation code fully match the parameters of the key file. If the parameters are different, the services are restarted.

KUMA generates an audit event after adding a license key, deleting a license key, or a license expiring.

When switching from a key file to an activation code, the old license key is deleted automatically. Before renewing your license, make sure that you have access to the old key file.

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 "solution") is an integrated software solution that combines the following functionality:

- Receiving, processing, and storing information security events
- Analyzing and correlating incoming data
- Searching in received 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 Kaspersky Security Center and KICS/KATA.
  - Create assets manually through the web interface or via the API.

  KUMA stores the following device information:

  - Technical characteristics of the device.
  - Vulnerabilities of the 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.
- Data received from GosSOPKA and messages from NCIRCC: information about the organization of the GosSOPKA subject, category and type of event, technical information about the malicious system (including domain name, IP addresses, hash, URI, email address), information about vulnerabilities of the attacked application and information about the attacked object.
- 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 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. Additionally, you can configure the saving of the address or host name of the event aggregator server.

- Information required for the integration of KUMA with other applications (Kaspersky Threat Lookup, Kaspersky CyberTrace, Kaspersky Security Center, Kaspersky Industrial CyberSecurity for Networks, Kaspersky Automated Security Awareness Platform, Kaspersky Endpoint Detection and Response, Security Orchestration, Automation and Response SOAR, AI services: AI score and asset status, Kaspersky Investigation & 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, an email address. 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. Assigned roles will be displayed.

  - 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.

- Information about current licenses in KUMA.

- 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 (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 collected data in anonymized form and only for general statistical purposes. Summary statistics is 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 is stored indefinitely.

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

Adding a license key

You can a license key in the KUMA web interface If you use a key file to activate the solution, two license keys, active and reserve, may be present. 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 **Available EPS** 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 *Manage* permission level for the *License* functional area can add a license key. Among the predefined roles, only the General Administrator has this permission.

How to add a license key

1. In the KUMA web interface, select the **Settings** → **Other** → **License** section.

   This opens a window with a list of added license groups.

2. Click **Add license key**.

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

4. Click **Add**.

   The new license group is displayed in the list.

5. Click the name of the added license group.

   This opens the license group configuration window.

6. Select the method for adding the license key:

   - Using an activation code:

     a. Enter the activation code.

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

   - Using a key file:

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

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

7. Click **Add**.

The license key is added.

How to replace the active license key

1. Delete the current license key.
2. Add a new license key (see instructions above).

How to add a reserve license key to a license group

1. In the KUMA web interface, select the **Settings** → **Other** → **License** section.

   This opens a window with a list of added license groups.
2. Click the name of the license group to which you want to add the reserve license key.
3. 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).
4. Click **Add**.

Example of using a merged license

This example shows the use of a merged license in a holding consisting of companies A, B, and C. One shared instance of KUMA is deployed and activated in the holding, and multitenancy is configured. It is impossible to configure multitenancy without activating the solution because activation is necessary to create services and resources.

Companies A, B, and C must each subsequently purchase a license of their own to use the solution.

To solve this problem, use a merged license:

1. Each of companies A, B and C purchases a license to use the solution in the form of a license certificate and a key file or an activation code.
2. Get key files and/or activation codes from companies A, B and C to activate the solution.
3. Use the key files and/or activation codes provided by companies A, B and C to add three active license keys (see the instructions above).

The license keys added during the deployment of the solution and at step 3 of the scenario are automatically merged. As a result, a merged license is created, the details of which are displayed in the **Settings** → **Other** → **License** section above the list of license groups.

Viewing information about the added license key

You can view information about the added license key in the KUMA web interface in the **Settings** → **Other** → **License** section.

Only users with the *Read* or *Manage* permissions for the *License* functional area can view license information. Among the predefined roles, only the General Administrator has this permission.

The **License** tab window displays the following information about added license keys:

- **Name of the license key**. Contains the name of the solution and additional parameters, if applicable.
- Unique alphanumeric sequence of the license key.
- **EPS available**—number of events processed per second supported by the license.
- **Current current per day** is the current average number of unique events processed over 24 hours, calculated using the sliding window method. Every 15 seconds, KUMA counts the events processed by the collector (after filtration and aggregation at the output of the collector) to get the average EPS per hour, and then per day.
- **EPS current per hour** is the current average amount of events processed by KUMA per hour.

  The **Current current per day** and **EPS current per hour** settings are available only for the SMART I and SMART II licenses. If values of these two settings are exceeded, the collector with the maximum EPS value stops receiving new events for one hour. After one hour, the collector resumes operation. Multiple collectors may be paused at the same time. A notification about the maximum EPS allowed by the license being exceeded is sent to the user with the General Administrator role. The limitations apply only to SMART I and SMART II licenses.

- **Expiration date** is the date when the license key will expire.
- **Days remaining** is the number of days before the license expires.
- **Modules** are modules covered by the license.

Deleting a license key

In KUMA, you can remove an added license key from the application (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 *Manage* permission level for the *License* functional area can delete a license key. Among the predefined roles, only the General Administrator has this permission.

To delete a license key, delete the license group to which the license key was added:

1. In the KUMA web interface, select the **Settings** → **Other** → **License** section.

   This opens a window with a list of added license groups.

2. Select the check box to the left of the name of the license group to select the license group that contains the license key that you want to delete.

3. Click the **Delete** button.

   A confirmation window opens.

4. Confirm the deletion of the license group.

The license key is removed from the solution.

Administrator's guide

This section provides information about installing and configuring the KUMA SIEM system.

Installing and removing KUMA

To install KUMA, you need the distribution kit:

- **kuma-ansible-installer-<build number>.tar.gz** contains all necessary files for installing KUMA without Kubernetes. You can perform a distributed installation of KUMA Core in a cluster based on the Raft algorithm (hereinafter referred to as the "Raft cluster").

- **kuma-ansible-installer-k8s-<build number>.tar.gz** contains all necessary files for installing KUMA in a high-availability configuration based on Kubernetes.

To complete the installation, you need the install.sh installer file and an inventory file that describes your infrastructure. You can create an inventory file based on a template. Each distribution contains an install.sh installer file and the following inventory file templates:

- single.inventory.yml.template

- distributed.inventory.yml.template

- expand.inventory.yml.template

- k0s.inventory.yml.template

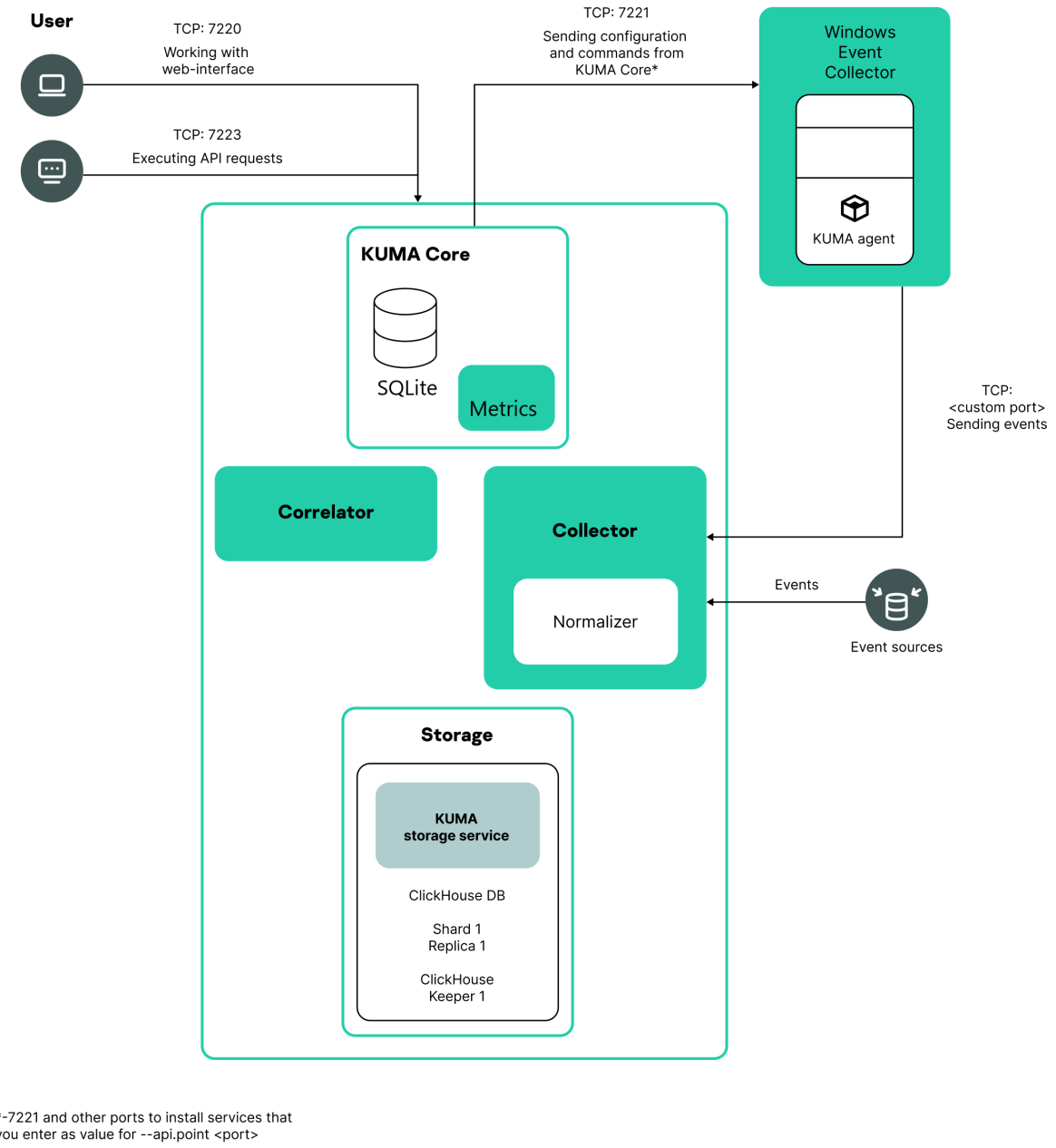
KUMA keeps its files in the /opt directory, so we recommend making /opt a separate partition and allocating 16 GB for the operating system and the remainder of the disk space for the /opt partition.

KUMA is installed in the same way on all hosts using the installer and your prepared inventory file in which you describe your configuration. We recommend taking time to think through the setup before you proceed.

The following installation options are available:

- Installation on a single server

Single-server installation diagram



Installation on a single server

Example inventory file for installation on a single server

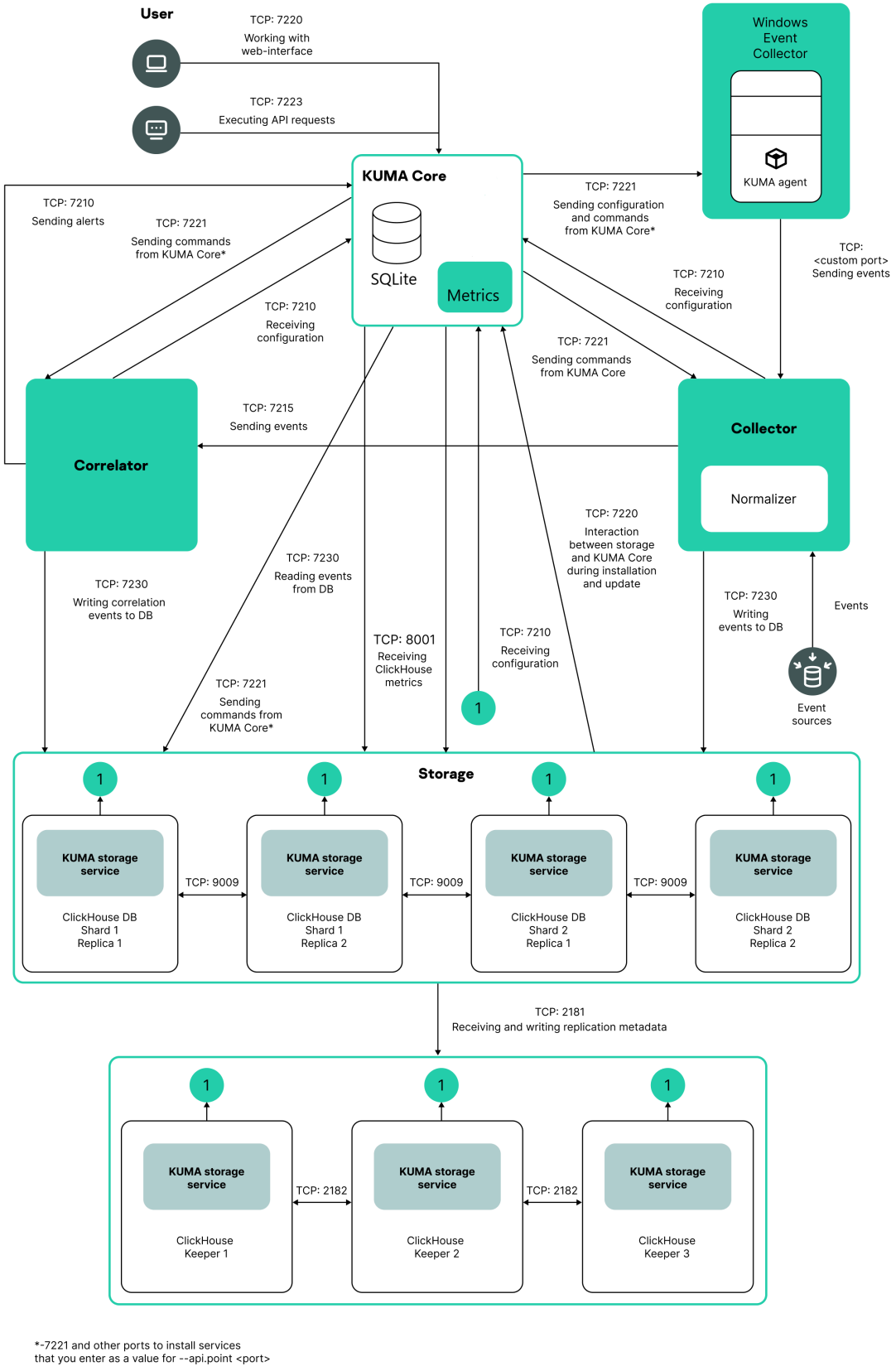
```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: true
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_ssh_private_key_file: <full path to the file with the SSH key>
    ansible_user: <user from the sudo group>
    ansible_become: true
    ansible_port: 2222
```

```
children:
  kuma_core:
    hosts:
      kuma1.example.com:
  kuma_collector:
    hosts:
      kuma1.example.com
  kuma_correlator:
    hosts:
      kuma1.example.com
  kuma_storage:
    hosts:
      kuma1.example.com:
        shard: 1
        replica: 1
        keeper: 1
  kuma_metrics:
    hosts:
      kuma1.example.com
```

You can install all KUMA components on the same server: specify the same server in the single.inventory.yml inventory file for all components. An "all-in-one" installation can handle a small stream of events, up to 10,000 EPS. If you plan to use many dashboard layouts and process a lot of search queries, a single server might not be sufficient. In that case, we recommend the distributed installation.

- Distributed installation

Distributed Installation diagram



Distributed installation diagram

Example inventory file for distributed installation

```
all:
```

```
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma-core-1.example.com:
          ip: 0.0.0.0
    kuma_collector:
      hosts:
        kuma-collector-1.example.com:
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
          ip: 0.0.0.0
    kuma_storage:
      hosts:
        kuma-storage-cluster1-server1.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server2.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server3.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server4.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server5.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 1
        kuma-storage-cluster1-server6.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 2
        kuma-storage-cluster1-server7.example.com:
          ip: 0.0.0.0
          shard: 0
```

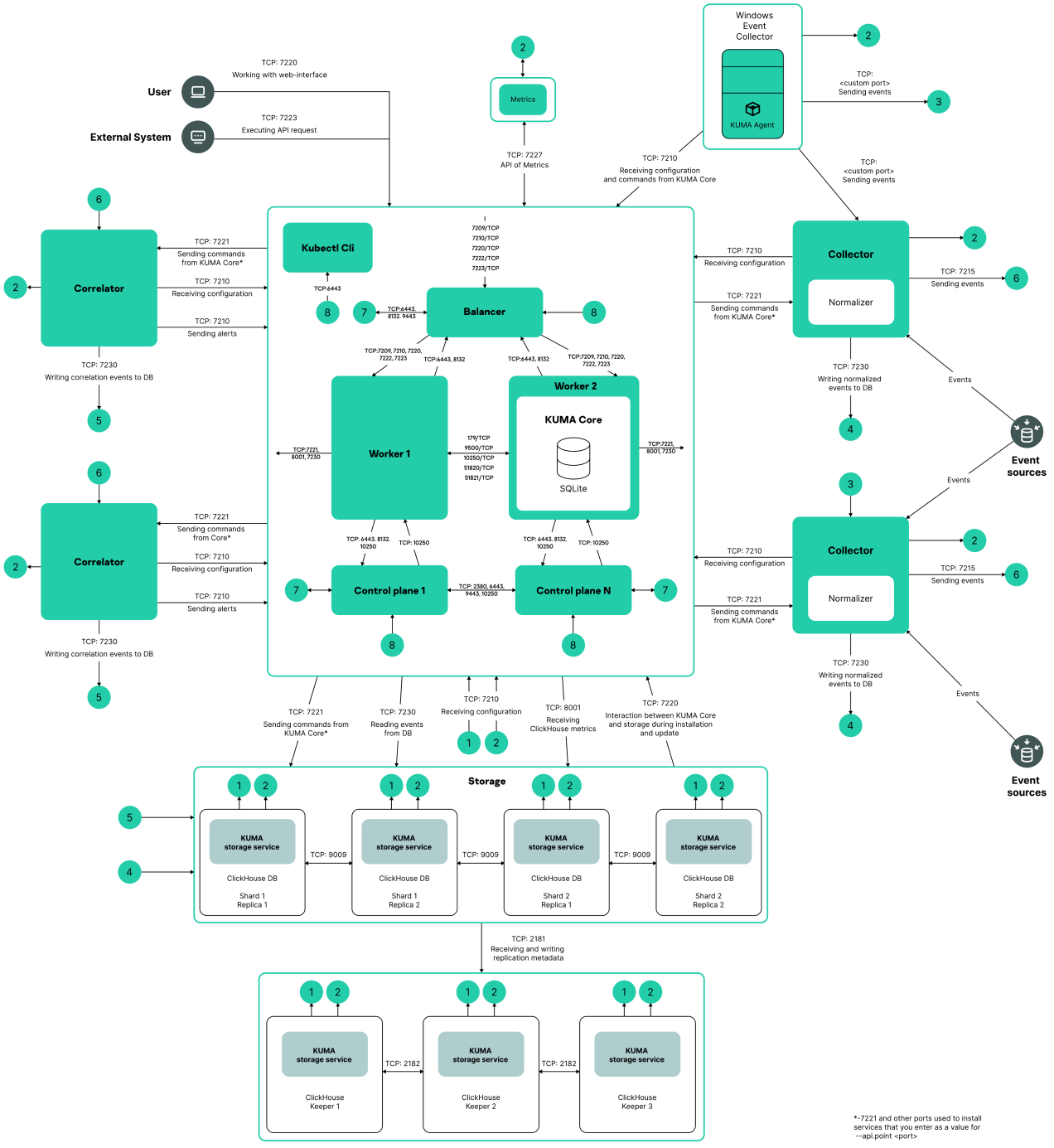
```
    replica: 0
    keeper: 3
```

You can install KUMA services on different servers; you can describe the configuration for a distributed installation in the distributed.inventory.yml inventory file.

- Installation in a high availability configuration

  - Distributed installation with KUMA Core in a Kubernetes cluster

    Distributed installation with KUMA Core in a Kubernetes cluster



Distributed installation in a high availability configuration

Example inventory file for distributed installation with KUMA Core in a Kubernetes cluster

```
all:
```

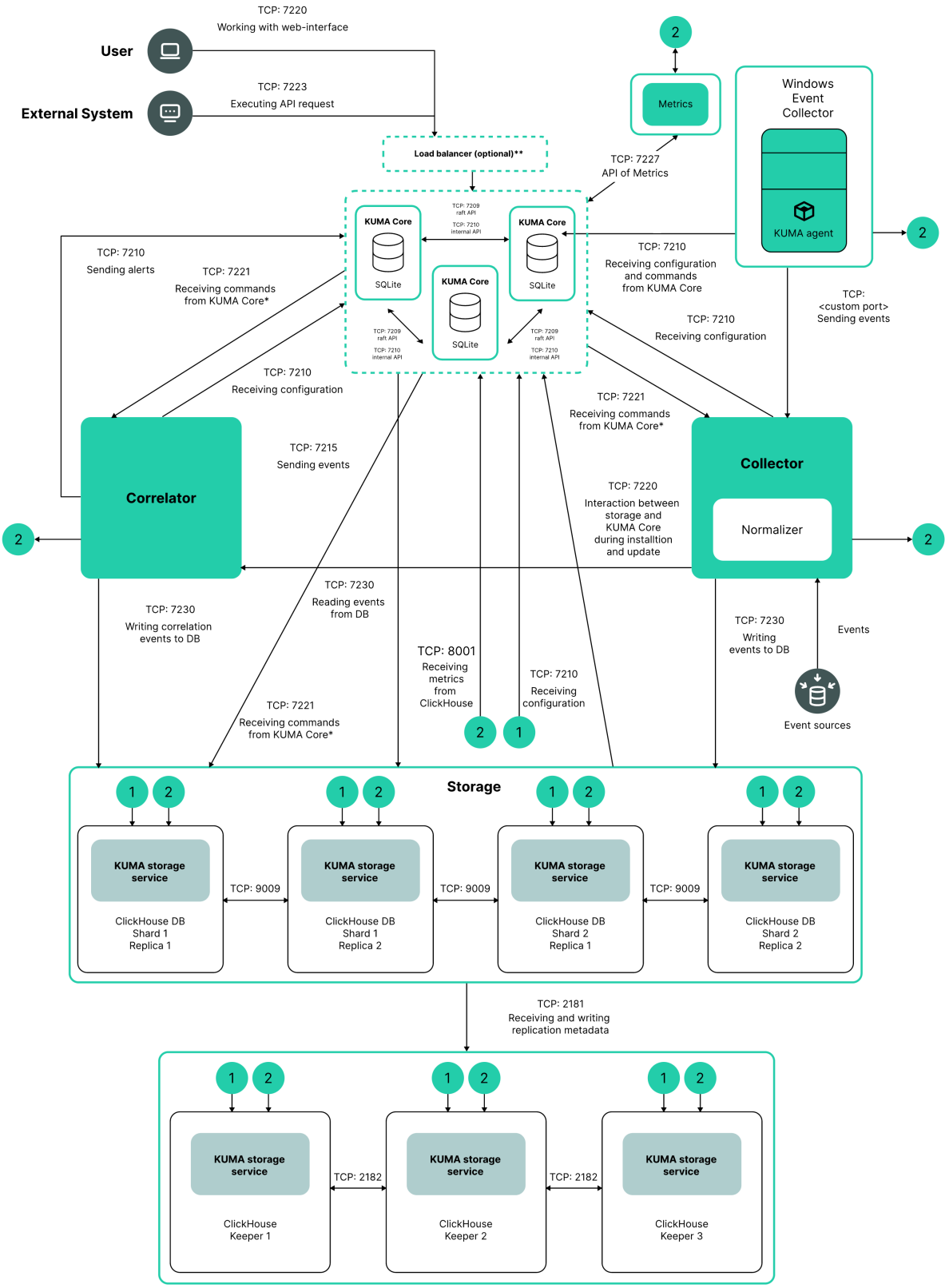
```
  vars:
    deploy_to_k8s: true
    need_transfer: true
    generate_etc_hosts: false
    airgap: true
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma-core-1.example.com:
    kuma_collector:
      hosts:
        kuma-collector-1.example.com:
          ip: 0.0.0.0
        kuma-collector-2.example.com:
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
          ip: 0.0.0.0
        kuma-correlator-2.example.com:
          ip: 0.0.0.0
    kuma_storage:
      hosts:
        kuma-storage-cluster1-server1.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server2.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server3.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server4.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server5.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 1
        kuma-storage-cluster1-server6.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
```

```
          keeper: 2
        kuma-storage-cluster1-server7.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 3
kuma_k0s:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_lb:
      hosts:
        kuma_lb.example.com:
          kuma_managed_lb: true
    kuma_control_plane_master:
      hosts:
        kuma_cpm.example.com:
          ansible_host: 10.0.1.10
    kuma_control_plane_master_worker:
    kuma_control_plane:
      hosts:
        kuma_cp1.example.com:
          ansible_host: 10.0.1.11
        kuma_cp2.example.com:
          ansible_host: 10.0.1.12
    kuma_control_plane_worker:
    kuma_worker:
      hosts:
        kuma-core-1.example.com:
          ansible_host: 10.0.1.13
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
        kuma_worker2.example.com:
          ansible_host: 10.0.1.14
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
```

You can install the KUMA Core on a Kubernetes cluster for high availability. Describe the configuration in the k0s.inventory.yml inventory file.

- Distributed installation with KUMA Core in a Raft cluster

Distributed installation with KUMA Core in a Raft cluster



*-7221 and other ports to install services that you enter as a value for --api.point <port>

**All KUMA services can address any of the available KUMA Core services directly, as shown on the diagram, or the services can interact with the cluster through a load balancer.

Example distributed.inventory.yml inventory file for initial installation

```
all:
```

```
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma-core-1.example.com:
          ip: 0.0.0.0
          raft_node_addr: ""
    kuma_core_peers:
      hosts:
        kuma-core-2.example.com:
          ip: 0.0.0.0
        kuma-core-3.example.com:
          ip: 0.0.0.0
    kuma_collector:
      hosts:
        kuma-collector-1.example.com:
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
          ip: 0.0.0.0
    kuma_storage:
      hosts:
        kuma-storage-cluster1-server1.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server2.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server3.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server4.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server5.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 1
        kuma-storage-cluster1-server6.example.com:
```

```
        ip: 0.0.0.0
        shard: 0
        replica: 0
        keeper: 2
      kuma-storage-cluster1-server7.example.com:
        ip: 0.0.0.0
        shard: 0
        replica: 0
        keeper: 3
```

Example inventory file for expanding the cluster using the expand.sh installer

```
kuma:
  vars:
    no_firewall_actions: false
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core_peers:
      hosts:
        kuma-core-4.example.com:
          ip: 0.0.0.0
        kuma-core-5.example.com:
          ip: 0.0.0.0
```

You can install multiple KUMA Cores in a Raft cluster for data replication and high availability in a distributed system. Describe the configuration in the distributed.inventory.yml inventory file.

Solution installation requirements

General solution installation requirements.

You can install the solution on the following operating systems:

- Oracle Linux
- Astra Linux
- Ubuntu
- RED OS
- MosOS

  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.

  RED OS 8 is supported without high availability (HA). When using RED OS 8 in the Server with GUI support configuration, you need to install the iscsi-initiator-utils package, and then run the following commands:

  ```
  systemctl enable iscsid
  systemctl start iscsid
  ```

Before deploying the solution, 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 installing the solution, make sure that the following command returns the correct FQDN of the host:

  ```
  hostname -f
  ```

- The server where the installer is run does not have the name `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.
- All hosts on which the KUMA services will be installed do not have a user named 'kuma'. This user is created automatically when installing KUMA services.

Installation requirements for Oracle Linux, Astra Linux, Ubuntu, RED OS, and MosOS devices

	Oracle Linux	Astra Linux	Ubuntu	RED OS	MosOS
Python version	3.6 to 3.11. Hosts with Python earlier than 3.6 (inclusive) cannot be combined in the same instance with Python 3.10 and later hosts.	3.6 to 3.11. Hosts with Python earlier than 3.6 (inclusive) cannot be combined in the same instance with Python 3.10 and later hosts.	3.6 to 3.11. Hosts with Python earlier than 3.6 (inclusive) cannot be combined in the same instance with Python 3.10 and later hosts.	3.6 to 3.11. Hosts with Python earlier than 3.6 (inclusive) cannot be combined in the same instance with Python 3.10 and later hosts.	3.6 to 3.11. Hosts with Python earlier than 3.6 (inclusive) cannot be combined in the same instance with Python 3.10 and later hosts.
SELinux module	Disabled	Disabled	Disabled	Disabled	Disabled

	Oracle Linux	Astra Linux	Ubuntu	RED OS	MosOS
Package manager	pip3	pip3	pip3	pip3	pip3

Basic packages	• netaddr
	• firewalld
	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`
		• python3-apt
		• curl
		• libcurl4
		• ufw
		To install the packages, run the following command:
		`apt install python3-apt curl libcurl4 ufw`
			• python3-apt
			• curl
			• libcurl4
			• ufw
			To install the packages, run the following command:
			`apt install python3-apt curl libcurl4 ufw`
				• netaddr
				• firewalld
				To install the packages, run the following commands:
				`pip3 install netaddr`
				`dnf install firewalld`
					• python3-rpm
					• firewalld
					To install the package, run the following commands:
					`zypper refresh`
					`zypper install python3-rpm firewalld`

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`	No value.	No value.
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 (for Ubuntu 22.04 LTS) or libasound2t64 (for Ubuntu 24.04 LTS). You can install these packages using the following command (for Ubuntu 22.04 LTS): `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` For Ubuntu 24.04 LTS, run the following command:	• 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: `dnf 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`	No value.

	Oracle Linux	Astra Linux	Ubuntu	RED OS	MosOS
			`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 libasound2t64`		
User permissions required to install the solution	No value.	You need to assign the required permissions to the user that will be installing the solution: `sudo pdpl-user -i 63 <user that will be installing the solution>`	No value.	No value.	No value.

Upgrading Oracle Linux 8.x to Oracle Linux 9.x with installed KUMA services

To upgrade from Oracle Linux 8.x to Oracle Linux 9.x:

1. Run the following commands to disable KUMA services on the hosts where the services are installed:
   - `sudo systemctl disable kuma-collector-<service ID>.service`
   - `sudo systemctl disable kuma-correlator-<service ID>.service`
   - `sudo systemctl disable kuma-storage-<service ID>.service`
   - `sudo systemctl disable kuma-metrics-<metrics service ID>.service`
   - `sudo systemctl disable kuma-vmalert.service`
   - `sudo systemctl disable kuma-core-<KUMA Core service ID>.service`
2. Upgrade the OS on every host.
3. Run the following commands to enable the services on the hosts where the services are installed:
   - `sudo systemctl enable kuma-core-<KUMA Core service ID>.service`
   - `sudo systemctl enable kuma-storage-<service ID>.service`
   - `sudo systemctl enable kuma-collector-<service ID>.service`
   - `sudo systemctl enable kuma-correlator-<service ID>.service`
   - `sudo systemctl enable kuma-metrics-<metrics service ID>.service`
   - `sudo systemctl enable kuma-vmalert.service`
4. Restart the hosts.

As a result, the upgrade is completed.

Ports used by KUMA during installation

For the solution to run correctly, you need to ensure that the KUMA components are able to interact with other components and applications over the network using the protocols and ports specified during the installation of the KUMA components.

Before installing KUMA, make sure that the following ports are free on the host from the `kuma_metrics` section in the inventory file, on which host you want to install the Metrics service:

- 7227
- 7228
- 8001

The table below lists the default ports. The installer automatically opens the ports during KUMA installation

Network ports used for the interaction of KUMA components

Protocol	Port	Direction	Purpose of the connection
HTTPS	7222	From the KUMA client to the KUMA Core server.	Reverse proxy to the CyberTrace system.
HTTPS	8123	Local requests from the storage service to the local node of the ClickHouse cluster.	Writing and getting normalized events in the ClickHouse cluster.
HTTPS	9009	Between replicas of the ClickHouse cluster.	Internal data communication between replicas of the ClickHouse cluster.
TCP	2181	From ClickHouse cluster nodes to the ClickHouse keeper replication coordination service.	Getting and writing replication metadata by replicas of ClickHouse servers.
TCP	2182	From one ClickHouse keeper replication coordination service to another.	Internal communication between replication coordination services to reach a quorum.
TCP	7210	From all KUMA components to the KUMA Core server.	Getting the KUMA configuration from the KUMA Core server, as well as recording the metrics of the KUMA agent.
TCP	7220	• From the KUMA client to the server with the KUMA Core component. • From storage hosts to the KUMA Core server during installation or upgrade.	• User access to the KUMA web interface. • Interaction between the storage hosts and the KUMA Core during installation or upgrade. You can close this port after completing the installation or upgrade.
TCP	7221 and other ports used for service installation as the --api.port <port> parameter value	From KUMA Core to KUMA services.	Administration of services from the KUMA web interface.
TCP	7223	To the KUMA Core server.	Default port for API requests.
TCP	8001	From the Metrics service to the ClickHouse server.	Getting ClickHouse server operation metrics.
TCP	9000	• Outgoing and incoming connections between servers of the ClickHouse cluster. • From the local client.sh client to the local cluster node.	Port of the ClickHouse native protocol (also called *ClickHouse TCP*). Used by ClickHouse applications and processes, such as clickhouse-server, clickhouse-client, and native ClickHouse tools Used for inter-server communication for distributed queries. Also used for writing and getting data in the ClickHouse cluster.

Ports used by predefined OOTB resources

The installer automatically opens the ports during KUMA installation.

Ports used by predefined OOTB resources:

- 7230/tcp
- 7231/tcp
- 7232/tcp
- 7233/tcp
- 7234/tcp
- 7235/tcp
- 5140/tcp
- 5140/udp
- 5141/tcp
- 5144/udp

KUMA Core traffic in a high availability configuration

The table below lists the initiator (source) of the connection and the destination. The port number on the initiator can be dynamic. Return traffic within the established connection must not be blocked.

KUMA Core traffic in a high availability configuration

Source	Destination	Destination port	Type
External KUMA services	Load balancer	7209	TCP
External KUMA services	Load balancer	7210	TCP
External KUMA services	Load balancer	7220	TCP
External KUMA services	Load balancer	7222	TCP
External KUMA services	Load balancer	7223	TCP
Worker node	Load balancer	6443	TCP
Worker node	Load balancer	8132	TCP
Control node	Load balancer	6443	TCP
Control node	Load balancer	8132	TCP
Control node	Load balancer	9443	TCP
Worker node	External KUMA services	Depending on the settings specified when creating the service.	TCP
Load balancer	Worker node	7209	TCP
Load balancer	Worker node	7210	TCP

Source	Destination	Destination port	Type
Load balancer	Worker node	7220	TCP
Load balancer	Worker node	7222	TCP
Load balancer	Worker node	7223	TCP
External KUMA services	Worker node	7209	TCP
External KUMA services	Worker node	7210	TCP
External KUMA services	Worker node	7220	TCP
External KUMA services	Worker node	7222	TCP
External KUMA services	Worker node	7223	TCP
Worker node	Worker node	179	TCP
Worker node	Worker node	9500	TCP
Worker node	Worker node	10250	TCP
Worker node	Worker node	51820	UDP
Worker node	Worker node	51821	UDP
Control node	Worker node	10250	TCP
Load balancer	Control node	6443	TCP
Load balancer	Control node	8132	TCP
Load balancer	Control node	9443	TCP
Worker node	Control node	6443	TCP
Worker node	Control node	8132	TCP
Worker node	Control node	10250	TCP
Control node	Control node	2380	TCP
Control node	Control node	6443	TCP
Control node	Control node	9443	TCP
Control node	Control node	10250	TCP
Cluster management console (CLI)	Load balancer	6443	TCP
Cluster management console (CLI)	Control node	6443	TCP

Downloading CA certificates

In the KUMA web interface, you can download the following CA certificates:

- REST API CA certificate.

  This certificate is used to authenticate the API server serving the KUMA public API. You can also use this certificate when importing data from MaxPatrol reports.

  You can also change this certificate if you want to use your company's certificate and key instead of the self-signed certificate of the web console.

  For an installation with the KUMA Core service in a Raft cluster without a balancer, you can download the CA certificate of the REST API to any node, but a secure connection is possible only with the core-1 node. The rest of the nodes return an error. If you want to directly log in with the same certificate to all nodes of the Raft cluster without a balancer, when generating a certificate, you can specify the FQDNs of all KUMA Core nodes in the Raft cluster in the Subject Alternative Names (SAN) field.

- Microservice CA certificate.

  This certificate is used for authentication when connecting log sources to passive collectors using TLS, but without specifying your own certificate.

To download a CA certificate:

1. Open the KUMA web interface.
2. In the lower left corner of the window, click the name of the user account, and in the menu, click the **REST API CA certificate** or **Microservice CA certificate** button, depending on the certificate that you want to download.

The certificate is saved to the download directory configured in your browser.

Reissuing internal CA certificates

Changed the storage location of the self-signed CA certificate and the certificate reissue mechanism.
The certificate is stored in the database. The previous method of reissuing internal certificates by deleting certificates from the file system of the Core and restarting the Core is no longer allowed. The old method will cause the Core to fail to start. Until the certificate reissuing process is completed, new services may not be connected to the Core.
After reissuing the internal CA certificates in the **Settings** → **Other** → **General** → **Reissue internal CA certificates** section of the KUMA web interface, you must stop the services, delete the old certificates from service directories, and manually restart all services. Only users with the General Administrator role can reissue internal CA certificates.

The **Reissue internal CA certificates** option is available only to a user with the General Administrator role.

The process of reissuing certificates for an individual service remains the same: in the KUMA web interface, in the **Resources** → **Active services** section, select the service; in the context menu, select **Reset certificate**, and delete the old certificate from the service installation directory. KUMA automatically generates a new certificate. You do not need to restart running services, the new certificate is applied automatically. A stopped service must be restarted to have the certificate applied.

To reissue internal CA certificates:

1. In the KUMA web interface, go to the **Settings** → **Other** → **General** section, click the **Reissue internal CA certificates** button, and review the displayed warning. If you decide to proceed with reissuing certificates, click **Yes**.

   As a result, the CA certificates for KUMA services and the CA certificate for ClickHouse are reissued. Next, you must stop the services, delete old certificates from the service installation directories, restart the Core, and restart the stopped services to apply the reissued certificates.

2. Connect to the hosts where the collector, correlator, and event router services are deployed.

   a. Stop all services with the following command:

   ```
   sudo systemctl stop kuma-<collector/correlator/eventRouter/metrics>-<service ID>.service
   ```

   b. Delete the internal.cert and internal.key certificate files from the "/opt/kaspersky/kuma/<service type>/<service ID>/certificates" directories with the following command:

```
sudo rm -f /opt/kaspersky/kuma/<service type>/<service ID>/certificates/internal.cert
sudo rm -f /opt/kaspersky/kuma/<service type>/<service ID>/certificates/internal.key
```

3. Connect to the hosts where storage services are deployed.

   a. Stop all storage services.

   ```
   sudo systemctl stop kuma-<storage>-<service ID>.service
   ```

   b. Delete the internal.cert and internal.key certificate files from the "/opt/kaspersky/kuma/storage/<ID service>/certificates" directories.

   ```
   sudo rm -f /opt/kaspersky/kuma/storage/<service ID>/certificates/internal.cert
   sudo rm -f /opt/kaspersky/kuma/storage/<service ID>/certificates/internal.key
   ```

4. Delete all ClickHouse certificates from the "/opt/kaspersky/kuma/clickhouse/certificates" directory.

   ```
   sudo rm -f /opt/kaspersky/kuma/clickhouse/certificates/internal.cert
   sudo rm -f /opt/kaspersky/kuma/clickhouse/certificates/internal.key
   ```

5. Connect to the hosts where agent services are deployed.

   a. Stop the services of Windows agents and Linux agents.

   b. Delete the internal.cert and internal.key certificate files from working directories of the agents.

6. Start the KUMA Core to apply the new CA certificates.

   - For an "all-in-one" or distributed installation of KUMA, run the following command:

     ```
     sudo systemctl restart kuma-core-00000000-0000-0000-0000-000000000000.service
     ```

   - For KUMA with a Core in a Kubernetes cluster, to restart the KUMA Core, run the following commands on any controller:

     ```
     sudo k0s kubectl exec deployment/metrics -n kuma -- rm -rf /opt/kaspersky/kuma/metrics/certificates
     sudo k0s kubectl rollout restart deployment metrics -n kuma
     sudo k0s kubectl rollout restart deployment/core-deployment -n kuma
     ```

     The KUMA Core must be restarted using the command because restarting the KUMA Core in the KUMA interface affects only the KUMA Core container and not the entire pod.

7. Restart all services that were stopped during the procedure.

   ```
   sudo systemctl start kuma-<collector/correlator/eventRouter/storage/metrics>-<service ID>.service
   ```

Internal CA certificates are reissued and applied.

Modifying the self-signed web console certificate

You can use your company's certificate and key instead of the self-signed certificate of the web console. For example, if you want to replace self-signed CA Core certificate with a certificate issued by an enterprise CA, you must provide an external.cert and an unencrypted external.key in PEM format.

The following example shows how to replace a self-signed CA Core certificate with an enterprise certificate in PFX format. You can use the instructions as an example and adapt the steps according to your needs.

To replace the certificate of the KUMA web console with an external certificate:

1. If you are using a certificate and key in a PFX container, use OpenSSL to convert the PFX file to a certificate and encrypted key in PEM format:

   ```
   openssl pkcs12 -in kumaWebIssuedByCorporateCA.pfx -nokeys -out external.cert
   openssl pkcs12 -in kumaWebIssuedByCorporateCA.pfx -nocerts -nodes -out external.key
   ```

   When carrying out the command, you are required to specify the PFX key password (Enter Import Password).

   As a result, the external.cert certificate and the external.key in PEM format are returned.

2. If your external.key is not encrypted, make sure that the key has the PKCS#8 format. KUMA will not accept a key file that begins with the BEGIN RSA PRIVATE KEY tag. In this case, convert the file using the following command:

   ```
   openssl rsa -in rsa_private.key -outform PEM -out private.key
   ```

   The file produced by this command begins with the BEGIN PRIVATE KEY tag, and KUMA accepts such a key file.

3. In the KUMA web interface, go to the **Settings** → **Other** → **General** → **Core settings** section. Under **External TLS pair**, click **Upload certificate** and **Upload key** and upload the external.cert file and the unencrypted external.key file in PEM format.

4. Restart KUMA:

   ```
   systemctl restart kuma-core
   ```

5. Refresh the web page or restart the browser hosting the KUMA web interface.

Your company certificate and key have been replaced.

Synchronizing time on servers

To configure time synchronization on servers:

1. Install chrony:

   ```
   sudo apt install chrony
   ```

2. Configure the synchronization of system time with an NTP server:

   a. Make sure the virtual machine has Internet access.

      If access is available, go to step b.

      If the virtual machine does not have internet access, edit the `/etc/chrony.conf` file to replace `2.pool.ntp.org` with the name or IP address of your corporate NTP server.

   b. Start the system time synchronization service:

      ```
      sudo systemctl enable --now chronyd
      ```

   c. Wait a few seconds and run the following command:

      ```
      sudo timedatectl | grep 'System clock synchronized'
      ```

      If the system time is synchronized correctly, the output will contain "System clock synchronized: yes".

Synchronization is configured.

About the inventory file

You can install, update, or remove KUMA components by changing to the directory with the extracted kuma-ansible-installer and using the Ansible tool and a prepared inventory file. You can specify KUMA configuration settings in the inventory file; the installer then uses these settings when deploying, updating, and removing the solution. The inventory file must conform to the YAML format.

You can create an inventory file based on the templates included in the distribution kit. The following templates are provided:

- single.inventory.yml.template can be used when installing KUMA on a single server. This template contains the minimum set of settings optimized for installation on a single device without using a Kubernetes cluster.
- distributed.inventory.yml.template can be used for the initial distributed installation of KUMA without using a Kubernetes cluster, for expanding an all-in-one installation to a distributed installation, and for updating KUMA.
- expand.inventory.yml.template can be used in some reconfiguration scenarios, such as adding collector and correlator servers, expanding an existing storage cluster, or adding a new storage cluster. If you use this inventory file to edit the configuration, the installer does not stop services in the entire infrastructure. If you reuse the inventory file, the installer can stop only services on hosts that are listed in the expand.inventory.yml inventory file.
- k0s.inventory.yml.template can be used to install or migrate KUMA to a Kubernetes cluster.

We recommend backing up the inventory file that you used to install the solution. You can use it to add components to the system or remove KUMA.

KUMA settings in the inventory file

The inventory file may include the following blocks:

- `all`
- `kuma`
- `kuma_k0s`

For each host, you must specify the FQDN in the `<host name>.<domain>` format and, if necessary, an IPv4 or IPv6 address. The KUMA Core domain name and its subdomains may not start with a numeral.

Example:

```
hosts:
  hostname.example.com:
    ip: 0.0.0.0
```

The 'all' block

In this block, you can specify the variables that apply to all hosts listed in the inventory file, including the implicitly specified localhost on which the installation is started. Variables can be overridden at the level of host groups or individual hosts.

Example of overriding variables in the inventory file

```
all:
  vars:
    ansible_connection: ssh
    deploy_to_k8s: False
    need_transfer: False
    airgap: True
    deploy_example_services: True
kuma:
  vars:
    ansible_become: true
    ansible_user: i.ivanov
    ansible_become_method: su
    ansible_ssh_private_key_file: ~/.ssh/id_rsa
  children:
    kuma_core:
      vars:
        ansible_user: p.petrov
        ansible_become_method: sudo
```

The table below lists all possible variables in the `vars` section and their descriptions.

List of possible variables in the vars section

Variable	Description
`ansible_connection`	Method used to connect to target computers. Possible values: • `ssh` to connect to remote hosts over SSH. If you are using a non-standard port to connect to remote hosts via SSH, specify the values for the `ansible_port` variable. • `local` to establish no connection with remote hosts.

Variable	Description

`ansible_port`	Variable for specifying a non-standard port for connecting to remote hosts via SSH. The `ansible_port` variable can be configured in the following ways:
	• If the same non-standard port is used on all hosts listed in the inventory file, specify the value of the `ansible_port` variable in the inventory file in the `all` block of the `vars` section.
	Example of a distributed.inventory.yml inventory file with the same non-standard port for all hosts
	`all:`
	`  vars:`
	`  ansible_port: 2222`
	`  deploy_to_k8s: false`
	`  need_transfer: false`
	`  generate_etc_hosts: false`
	`  deploy_example_services: false`
	`  no_firewall_actions: false`
	`kuma:`
	`  vars:`
	`  ansible_connection: ssh`
	`  ansible_user: root`
	`  children:`
	`  kuma_core:`
	`  hosts:`
	`  kuma-core-1.example.com:`
	`  ip: 0.0.0.0`
	`  kuma_collector:`
	`  hosts:`
	`  kuma-collector-1.example.com:`
	`  ip: 0.0.0.0`
	`  kuma_correlator:`
	`  hosts:`
	`  kuma-correlator-1.example.com:`
	`  ip: 0.0.0.0`
	`  kuma_storage:`
	`  hosts:`
	`  kuma-storage-cluster1-server1.example.com:`
	`  ip: 0.0.0.0`
	`  shard: 1`
	`  replica: 1`
	`  keeper: 0`
	`  kuma-storage-cluster1-server2.example.com:`
	`  ip: 0.0.0.0`
	`  shard: 1`
	`  replica: 2`
	`  keeper: 0`
	`  kuma-storage-cluster1-server3.example.com:`
	`  ip: 0.0.0.0`
	`  shard: 2`
	`  replica: 1`
	`  keeper: 0`
	`  kuma-storage-cluster1-server4.example.com:`
	`  ip: 0.0.0.0`
	`  shard: 2`
	`  replica: 2`
	`  keeper: 0`
	`  kuma-storage-cluster1-server5.example.com:`
	`  ip: 0.0.0.0`
	`  shard: 0`
	`  replica: 0`
	`  keeper: 1`
	`  kuma-storage-cluster1-server6.example.com:`
	`  ip: 0.0.0.0`
	`  shard: 0`
	`  replica: 0`
	`  keeper: 2`
	`  kuma-storage-cluster1-server7.example.com:`
	`  ip: 0.0.0.0`
	`  shard: 0`
	`  replica: 0`
	`  keeper: 3`
	• If you are using a non-standard port for a group of hosts, specify the value of the variable for the group of hosts in the inventory file in the `kuma` or `kuma_k0s` block in the `vars` section. For example:
	Example of a distributed.inventory.yml inventory file with the same non-standard port for a group of storage hosts
	`all:`
	`  vars:`
	`  deploy_to_k8s: false`
	`  need_transfer: false`
	`  generate_etc_hosts: false`
	`  deploy_example_services: false`
	`  no_firewall_actions: false`
	`kuma:`
	`  vars:`

Variable	Description
	``` ansible_connection: ssh ansible_user: root children: kuma_core: hosts: kuma-core-1.example.com: ip: 0.0.0.0 kuma_collector: hosts: kuma-collector-1.example.com: ip: 0.0.0.0 kuma_correlator: hosts: kuma-correlator-1.example.com: ip: 0.0.0.0 kuma_storage: vars: ansible_port: 2222 hosts: kuma-storage-cluster1-server1.example.com: ip: 0.0.0.0 shard: 1 replica: 1 keeper: 0 kuma-storage-cluster1-server2.example.com: ip: 0.0.0.0 shard: 1 replica: 2 keeper: 0 kuma-storage-cluster1-server3.example.com: ip: 0.0.0.0 shard: 2 replica: 1 keeper: 0 kuma-storage-cluster1-server4.example.com: ip: 0.0.0.0 shard: 2 replica: 2 keeper: 0 kuma-storage-cluster1-server5.example.com: ip: 0.0.0.0 shard: 0 replica: 0 keeper: 1 kuma-storage-cluster1-server6.example.com: ip: 0.0.0.0 shard: 0 replica: 0 keeper: 2 kuma-storage-cluster1-server7.example.com: ip: 0.0.0.0 shard: 0 replica: 0 keeper: 3 ``` **Example of a k0s.inventory.yml inventory file with the same non-standard port for all hosts in the kuma_control_plane section** ``` all: vars: ansible_connection: ssh ansible_user: root deploy_to_k8s: true need_transfer: false generate_etc_hosts: false deploy_example_services: false no_firewall_actions: false metrics_disk_request: 150Gi core_disk_request: 100Gi kuma: children: kuma_collector: hosts: kuma-collector.example.com: kuma_correlator: hosts: kuma-correlator.example.com: kuma_storage: hosts: kuma-storage-cluster1.server1.example.com kuma-storage-cluster1.server2.example.com kuma-storage-cluster1.server3.example.com kuma-storage-cluster1.server4.example.com kuma-storage-cluster1.server5.example.com kuma-storage-cluster1.server6.example.com ```

Variable	Description
	kuma-storage-cluster1.server7.example.com kuma_k0s: children: kuma_lb: hosts: kuma-lb.example.com: kuma_managed_lb: true kuma_control_plane_master: hosts: kuma_cpm.example.com: ansible_host: 10.0.1.10 kuma_control_plane_master_worker: kuma_control_plane: vars: ansible_port: 2222 hosts: kuma_cp2.example.com: ansible_host: 10.0.1.11 kuma_cp3.example.com: ansible_host: 10.0.1.12 kuma_control_plane_worker: kuma_worker: hosts: kuma-w1.example.com: ansible_host: 10.0.2.11 extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-ingress=true,node.longhorn.io/create-default-disk=true" kuma-w2.example.com: ansible_host: 10.0.2.12 extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-ingress=true,node.longhorn.io/create-default-disk=true" **Example of a distributed.inventory.yml inventory file for a high-availability installation with KUMA Core in a Raft cluster with the same non-standard port for all KUMA Core hosts** all: vars: deploy_to_k8s: false need_transfer: false generate_etc_hosts: false deploy_example_services: false no_firewall_actions: false kuma: vars: ansible_connection: ssh ansible_user: root children: kuma_core: hosts: kuma-core-1.example.com: ip: 0.0.0.0 raft_node_addr: "" ansible_port: 2222 kuma_core_peers: vars: ansible_port: 2222 hosts: kuma-core-2.example.com: ip: 0.0.0.0 kuma-core-3.example.com: ip: 0.0.0.0 kuma_collector: hosts: kuma-collector-1.example.com: ip: 0.0.0.0 kuma_correlator: hosts: kuma-correlator-1.example.com: ip: 0.0.0.0 kuma_storage: hosts: kuma-storage-cluster1-server1.example.com: ip: 0.0.0.0 shard: 1 replica: 1 keeper: 0 kuma-storage-cluster1-server2.example.com: ip: 0.0.0.0 shard: 1 replica: 2 keeper: 0 kuma-storage-cluster1-server3.example.com: ip: 0.0.0.0 shard: 2 replica: 1

Variable	Description

	keeper: 0
	kuma-storage-cluster1-server4.example.com:
	ip: 0.0.0.0
	shard: 2
	replica: 2
	keeper: 0
	kuma-storage-cluster1-server5.example.com:
	ip: 0.0.0.0
	shard: 0
	replica: 0
	keeper: 1
	kuma-storage-cluster1-server6.example.com:
	ip: 0.0.0.0
	shard: 0
	replica: 0
	keeper: 2
	kuma-storage-cluster1-server7.example.com:
	ip: 0.0.0.0
	shard: 0
	replica: 0
	keeper: 3
	• If you are using a different non-standard SSH connection port for each host, specify the value of the ansible_port variable for each host in the inventory file.
	Example of a distributed.inventory.yml inventory file with a different non-standard port for each storage host
	`all:`
	`vars:`
	`deploy_to_k8s: false`
	`need_transfer: false`
	`generate_etc_hosts: false`
	`deploy_example_services: false`
	`no_firewall_actions: false`
	`kuma:`
	`vars:`
	`ansible_connection: ssh`
	`ansible_ssh_private_key_file: <full path to the file with the SSH key>`
	`ansible_user: <user from the sudo group>`
	`ansible_become: true`
	`children:`
	`kuma_core:`
	`hosts:`
	`kuma-core-1.example.com:`
	`raft_node_addr: ""`
	`ip: 0.0.0.0`
	`kuma_metrics:`
	`hosts:`
	`kuma-metrics.example.com`
	`ip: 0.0.0.0`
	`kuma_collector:`
	`hosts:`
	`kuma-collector-1.example.com:`
	`ansible_port: 2222`
	`ip: 0.0.0.0`
	`kuma_correlator:`
	`hosts:`
	`kuma-correlator-1.example.com:`
	`ansible_port: 2223`
	`ip: 0.0.0.0`
	`kuma_storage:`
	`vars:`
	`ansible_port: 2224`
	`hosts:`
	`kuma-storage-cluster1-server1.example.com:`
	`ip: 0.0.0.0`
	`shard: 1`
	`replica: 1`
	`keeper: 0`
	`kuma-storage-cluster1-server2.example.com:`
	`ip: 0.0.0.0`
	`shard: 1`
	`replica: 2`
	`keeper: 0`
	`kuma-storage-cluster1-server3.example.com:`
	`ip: 0.0.0.0`
	`shard: 2`
	`replica: 1`
	`keeper: 0`
	`kuma-storage-cluster1-server4.example.com:`
	`ip: 0.0.0.0`
	`shard: 2`
	`replica: 2`
	`keeper: 0`
	`kuma-storage-cluster1-server5.example.com:`
	`ip: 0.0.0.0`
	`shard: 0`
	`replica: 0`
	`keeper: 1`
	`kuma-storage-cluster1-server6.example.com:`
	`ip: 0.0.0.0`
	`shard: 0`
	`replica: 0`
	`keeper: 2`
	`kuma-storage-cluster1-server7.example.com:`
	`ip: 0.0.0.0`
	`shard: 0`
	`replica: 0`
	`keeper: 3`
	Example of a k0s.inventory.yml inventory file with a different non-standard port for each hosts in the kuma_control_plane group
	`all:`
	`vars:`
	`ansible_connection: ssh`

Variable	Description
	<pre> ansible_user: root deploy_to_k8s: true need_transfer: false generate_etc_hosts: false deploy_example_services: false no_firewall_actions: false metrics_disk_request: 150Gi core_disk_request: 100Gi kuma: children: kuma_collector: hosts: kuma-collector.example.com: kuma_correlator: hosts: kuma-correlator.example.com: kuma_storage: hosts: kuma-storage-cluster1.server1.example.com kuma-storage-cluster1.server2.example.com kuma-storage-cluster1.server3.example.com kuma-storage-cluster1.server4.example.com kuma-storage-cluster1.server5.example.com kuma-storage-cluster1.server6.example.com kuma-storage-cluster1.server7.example.com kuma_k0s: children: kuma_lb: hosts: kuma-lb.example.com: kuma_managed_lb: true kuma_control_plane_master: hosts: kuma_cpm.example.com: ansible_host: 10.0.1.10 kuma_control_plane_master_worker: kuma_control_plane: hosts: kuma_cp2.example.com: ansible_port: 2222 ansible_host: 10.0.1.11 kuma_cp3.example.com: ansible_port: 2223 ansible_host: 10.0.1.12 kuma_control_plane_worker: kuma_worker: hosts: kuma-w1.example.com: ansible_host: 10.0.2.11 extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma- ingress=true,node.longhorn.io/create-default-disk=true" kuma-w2.example.com: ansible_host: 10.0.2.12 extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma- ingress=true,node.longhorn.io/create-default-disk=true"</pre> **Example of a distributed.inventory.yml inventory file for a high-availability installation with KUMA Core in a Raft cluster with a different non-standard port for each KUMA Core host** <pre>all: vars: deploy_to_k8s: false need_transfer: false generate_etc_hosts: false deploy_example_services: false no_firewall_actions: false kuma: vars: ansible_connection: ssh ansible_user: root children: kuma_core: hosts: kuma-core-1.example.com: ip: 0.0.0.0 raft_node_addr: "" ansible_port: 2222 kuma_core_peers: hosts: kuma-core-2.example.com: ip: 0.0.0.0 ansible_port: 2223 kuma-core-3.example.com: ip: 0.0.0.0 ansible_port: 2224</pre>

Variable	Description
	<pre>kuma_collector: hosts: kuma-collector-1.example.com: ip: 0.0.0.0 kuma_correlator: hosts: kuma-correlator-1.example.com: ip: 0.0.0.0 kuma_storage: hosts: kuma-storage-cluster1-server1.example.com: ip: 0.0.0.0 shard: 1 replica: 1 keeper: 0 kuma-storage-cluster1-server2.example.com: ip: 0.0.0.0 shard: 1 replica: 2 keeper: 0 kuma-storage-cluster1-server3.example.com: ip: 0.0.0.0 shard: 2 replica: 1 keeper: 0 kuma-storage-cluster1-server4.example.com: ip: 0.0.0.0 shard: 2 replica: 2 keeper: 0 kuma-storage-cluster1-server5.example.com: ip: 0.0.0.0 shard: 0 replica: 0 keeper: 1 kuma-storage-cluster1-server6.example.com: ip: 0.0.0.0 shard: 0 replica: 0 keeper: 2 kuma-storage-cluster1-server7.example.com: ip: 0.0.0.0 shard: 0 replica: 0 keeper: 3</pre>
`ansible_user`	User name used to connect to target computers and install components. If root login is blocked on the target computers, choose a user that has the right to establish SSH connections and elevate privileges using su or sudo.
`ansible_become`	This variable specifies if you want to elevate the privileges of the user that is used to install KUMA components. Possible values: • You must specify `true` if `ansible_user` is not `root`. • `false`.
`ansible_become_method`	Method for elevating the privileges of the user that is used to install KUMA components. You must specify `su` or `sudo` if `ansible_user` is not `root`.
`ansible_ssh_private_key_file`	Path to the private key in the /<path>/.ssh/id_rsa format. You must specify this variable if you want to use a key file other than the default key file (~/.ssh/id_rsa).
`deploy_to_k8s`	This variable specifies whether you want to deploy KUMA components in a Kubernetes cluster. Possible values: • The `false` value is specified in the single.inventory.yml and distributed.inventory.yml templates. • The `true` value is specified in the k0s.inventory.yml template. If you do not specify this variable, it defaults to `false`.
`need_transfer`	This variable specifies whether you want to migrate KUMA Core to a new Kubernetes cluster. You need to specify this variable only if `deploy_to_k8s` is `true`. Possible values: • `false` means KUMA must be installed or updated on the Kubernetes cluster. • `true` means KUMA Core must be migrated to a new Kubernetes cluster. If you do not specify this variable, it defaults to `false`.

Variable	Description
`no_firewall_actions`	This variable specifies whether the installer must perform the steps to configure the firewall on the hosts. Possible values: • `true` means that at startup, the installer does not perform the steps to configure the firewall on the hosts. • `false` means that at startup, the installer performs the steps to configure the firewall on the hosts. This is the value that is specified in all inventory file templates. If you do not specify this variable, it defaults to `false`.
`generate_etc_hosts`	This variable specifies whether the computers must be registered in the DNS zone of your organization. The installer automatically adds the IP addresses of the computers from the inventory file to the /etc/hosts files on the computers on which KUMA components are installed. The specified IP addresses must be unique. Possible values: • `false`. • `true`. If you do not specify this variable, it defaults to `false`. When using `generate_etc_hosts: true` in the inventory file for a high-availability configuration, you must specify the balancer IP address in the `ansible_host` variable for the balancer host in the `kuma_lb` section.
`deploy_example_services`	This variable specifies whether predefined services are created during the installation of KUMA. You need to specify this variable if you want to create demo services independently of the single/distributed/k0s inventory file. Possible values: • `false` means predefined services are not created when installing KUMA. This is the value that is specified in all inventory file templates. • `true` means predefined services are created when installing KUMA. If you do not specify this variable, it defaults to `false`.
`low_resources`	This variable specifies whether KUMA is being installed in an environment with limited computational resources. This variable is not specified in any of the inventory file templates. Possible values: • `false` means KUMA is being installed for production use. In this case, the installer checks the requirements of the worker nodes (CPU, RAM, and free disk space) in accordance with the hardware and software requirements. If the requirements are not satisfied, the installation is aborted with an error message. • `true` means that KUMA is being installed in an environment with limited computational resources. In this case, the minimum size of the KUMA Core installation directory on the host is 4 GB. All other computational resource limitations are ignored. If you do not specify this variable, it defaults to `false`.
`core_disk_request`	This variable sets the size of the KUMA Core volume for an installation in a Kubernetes cluster. If this variable is not set, the following default values are used for the KUMA Core volume size during installation: • If `low_resources=false`, the default volume size is 100Gi. • If `low_resources=true`, the default volume size is 4Gi. The 'Gi' suffix is required.
`metrics_disk_request`	This variable sets the size of the Metrics service volume for an installation in a Kubernetes cluster. If this variable is not set, the following default values are used for the Metrics server volume size during installation: • If `low_resources=false`, the default volume size is 80Gi. • If `low_resources=true`, the default volume size is 4Gi. The 'Gi' suffix is required.
`metrics_transfer_job_timeout`	This variable sets the timeout in minutes for copying metrics from a KUMA 3.4 Core volume to a new metrics volume in KUMA 4.0. The default setting is 120m. The 'm' suffix is required.
`mtu`	This variable allows you to set the maximum size of data that can be transmitted in one packet over the network medium in the Kubernetes cluster without the need for fragmentation. To solve the problem with excessive retransmission of packets or complete loss of connectivity in the encrypted link, we recommend specifying an `mtu` less than the minimum `mtu` of the existing link between the worker nodes. If no value is specified, the size of data is automatically determined based on the types of host interfaces.

Variable	Description
`backup_timeout`	This variable allows setting a maximum time in seconds allotted for creating a backup copy of the KUMA Core and separately for restoring data from a backup copy in the Kubernetes cluster. For example, if you set this variable to 2700, it means that 2700 seconds is allocated for the creation of a backup copy and 2700 seconds for restoration from a backup copy. If this setting is not specified in the 'vars' section of the inventory file, the default value is applied when automatically creating a backup copy of the KUMA Core or automatically restoring data from backup. The default value is 1800 seconds. If the automatic backup is completed before the specified time elapses, the upgrade script proceeds with the next step immediately. It makes sense to change this setting if manually creating a backup copy of the KUMA Core takes close to 30 minutes or more.

The 'kuma' block

In this block, you can specify the settings of KUMA components deployed outside of the Kubernetes cluster. The `kuma` block can contain the following sections:

- `vars` contains variables that apply to all hosts specified in the `kuma` block.
- `children` contains groups of settings for components:
  - `kuma_core` contains settings of the KUMA Core.
    - `raft_node_addr` is the address to be used for the cluster of nodes with the KUMA Core service. This address must be public and known to other nodes. After the KUMA Core is started for the first time with this address, the address cannot be changed anymore for this node. This value must be specified in the <FQDN>:<`port`> format. If this setting is not specified explicitly, <FQDN> defaults to the FQDN of the host on which the KUMA Core is deployed, and <`port`> defaults to 7209. The `raft_node_addr` setting must be specified together with the `raft_node_listen setting on each KUMA Core server with if DNS is configured in such a way that the server contains multiple FQDNs`. Optional setting.
    - `raft_node_listen` is the address at which the node listens for messages from its peers in the cluster. This setting must be specified in the following format <`IPv4 or IPv6 address of the KUMA Core server`>:<`port`>. This address can be changed. The `raft_node_listen` setting must be specified together with the `raft_node_addr` setting on every KUMA Core server. Optional setting.
  - `kuma_core_peers` contains settings of additional KUMA Cores. The installer prepares the servers specified in the `kuma_core_peers` group for the subsequent installation of additional KUMA Core services. You must specify an even number of servers in this group. If the `kuma_core_peers` group is missing from the inventory file or is empty, the normal installation or upgrade procedure with a single KUMA Core is performed. Only one KUMA Core service can be installed on each server in the `kuma_core_peers` group.

    Using the kuma_core_peers group for different installer use cases

Installer use case	The inventory file contains a kuma_core_peers group	The inventory file does not contain a kuma_core_peers group or the group is empty
Installation using the install.sh installer	Servers from the group are prepared for running the KUMA Core service. The kuma-core-00000000-0000-0000-0000-000000000000 service is installed on a server from the kuma_core group (displayed as core-1 in the web interface). On all servers from the kuma_core_peers group, the kuma-core-UID services are sequentially installed and started (displayed as core-<FQDN> in the web interface). These services are connected to the core-1 service. All additional services also connect to mongoDB running on the core-1 host, so the mongoDB is bound to all IP addresses of the core-1 server, and port 27017/tcp is allowed on the firewall.	A standard installation is performed with one core-1 service and one metrics service on a server from the kuma_core group.
Upgrading from version 3.4.x using the install.sh installer	Servers from the kuma_core_peersgroup are prepared for running the KUMA Core service. KUMA Core services are not installed on these servers. You can later create additional KUMA Core services in the web interface and install these on the prepared servers.	Normal upgrade is performed, the old metrics services are replaced by the new servers running on the server from the kuma_core group.
Expanding the installation using the expand.sh installer	You can create or expand a Raft cluster in an existing installation. To do this, you need to create an expand.inventory.yml inventory file, but fill in only the kuma_core_peers group. After running the expand.sh installer with the expand.inventory.yml inventory file, you need to manually create the KUMA Core services in the web interface and install them on the servers prepared by the installer. After expanding the Raft cluster, we recommend reflecting the changes in the kuma_core_peers group in the inventory file distributed.inventory.yml in case you need to upgrade KUMA.	A normal installation expansion is performed.

- `raft_node_addr` is the address to be used for the cluster of nodes with the KUMA Core service. This address must be public and known to other nodes. After the KUMA Core is started for the first time with this address, the address cannot be changed anymore for this node. This value must be specified in the <FQDN>:`<port>` format. If this setting is not specified explicitly, <FQDN> defaults to the FQDN of the host on which the KUMA Core is deployed, and `<port>` defaults to 7209. The `raft_node_addr` setting must be specified together with the `raft_node_listen setting on each KUMA Core server with if DNS is configured in such a way that the server contains multiple FQDNs`. Optional setting.

- `raft_node_listen` is the address at which the node listens for messages from its peers in the cluster. This setting must be specified in the following format `<IPv4 or IPv6 address of the KUMA Core server>:<port>`. This address can be changed. The `raft_node_listen` setting must be specified together with the `raft_node_addr` setting on every KUMA Core server. Optional setting.

- `kuma_metrics` contains settings of the Metrics service. You can only specify one host. 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.

- `kuma_collector` contains settings of KUMA collectors. You can specify multiple hosts.

- `kuma_correlator` contains settings of KUMA correlators. You can specify multiple hosts.

- `kuma_storage` contains settings of KUMA storage nodes. You can specify multiple hosts as well as shard, replica, and keeper IDs for hosts using the following settings:

  - `shard` is the shard ID.

  - `replica` is the replica ID.

  - `keeper` is the keeper ID.

The specified shard, replica, and keeper IDs are used only if you are deploying demo services as part of a fresh KUMA installation. In other cases, the shard, replica, and keeper IDs that you specified in the KUMA web interface when creating a resource set for the storage are used.

The 'kuma_k0s' block

In this block, you can specify the settings of the Kubernetes cluster that ensures high availability of KUMA. This block is specified only in an inventory file based on k0s.inventory.yml.template.

For test and demo installations in environments with limited computational resources, you must also set `low_resources: true` in the `all` block. In this case, the minimum size of the KUMA Core installation directory is reduced to 4 GB and the limitations of other computational resources are ignored.

For each host in the `kuma_k0s` block, a unique FQDN and IP address must be specified in the `ansible_host` variable, except for the host in the `kuma_lb` section. For the host in the `kuma_lb` section, only the FQDN must be specified. Hosts must be unique within a group.

For a demo installation, you may combine a controller with a worker node. Such a configuration does not provide high availability of the KUMA Core and is only intended for demonstrating the functionality or for testing the software environment.
The minimal configuration that ensures high availability is 3 controllers, 2 worker nodes, and 1 nginx load balancer. In production, we recommend using dedicated worker nodes and controllers. If a cluster controller is under workload and the pod with the KUMA Core is hosted on the controller, if the controller goes down, access to the KUMA Core will be completely lost.

The `kuma_k0s` block can contain the following sections:

- `vars` contains variables that apply to all hosts specified in the `kuma` block.
- `children` contains settings of the Kubernetes cluster that provides high availability of KUMA.

The following table lists possible variables in the `vars` section and their descriptions.

List of possible variables in the `vars` section

Variable group	Description	
`kuma_lb`	FQDN of the load balancer. You can install the nginx load balancer or a third-party TCP load balancer. If you are installing the nginx load balancer, you can set `kuma_managed_lb=true` to automatically configure the nginx load balancer when installing KUMA, open the necessary network ports on the nginx load balancer host (6443, 8132, 9443, 7209, 7210, 7220, 7222, 7223, 7226), and restart to apply the changes. If you are installing a third-party TCP load balancer, you must manually configure it before installing KUMA.	
`kuma_control_plane_master`	The host that acts as the primary controller of the cluster.	Groups for specifying the primary controller. You only need to specify a host for one group.
`kuma_control_plane_master_worker`	A host that combines the role of the primary controller and worker node in the cluster. For each cluster controller that is combined with a worker node, in the inventory file, you must specify `extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-ingress=true,node.longhorn.io/create-default-disk=true"`.	

Variable group	Description	
`kuma_control_plane`	Hosts that act as controllers in the cluster.	Groups for specifying secondary controllers.
`kuma_control_plane_worker`	Hosts that combine the roles of controller and worker node in the cluster. For each cluster controller that is combined with a worker node, in the inventory file, you must specify `extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-ingress=true,node.longhorn.io/create-default-disk=true"`.	
`kuma_worker`	Worker nodes of the cluster. For each cluster controller that is combined with a worker node, in the inventory file, you must specify `extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-ingress=true,node.longhorn.io/create-default-disk=true"`.	
`kuma_metrics`	Settings of the Metrics service. You can specify only one KUMA Core host. When migrating the KUMA Core from a distributed installation 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.	

`ipAutodetectionMethod`	If multiple network interfaces are being used on the worker nodes of the cluster at the same time, the `ipAutodetectionMethod` variable lets you specify a name mask of the network interface to be used for communication between the worker nodes in the cluster.
	For example, if you want to use only network interfaces named ethN (where N is the number of the network interface) for communication between worker nodes of the cluster, you can specify the variable as follows:
	`kuma_k0s:`
	`vars:`
	`ip_autodetection_method: "interface=eth.*"`
	This makes the cluster use a network interface with a name that matches the `eth.*` mask.
	If the network interface name on each worker node is the same, for example eth0, you can specify the variable without a mask:
	`kuma_k0s:`
	`vars:`
	`ip_autodetection_method: "interface=eth0"`
	For more information, please refer to the Calico Open Source documentation.	

Installation on a single server

To install KUMA components on a single server, complete the following steps:

1. Make sure that hardware, software, and installation requirements for KUMA are met.

2. Prepare the single.inventory.yml inventory file.

   Use the single.yml.template inventory file template from the distribution kit to create a single.inventory.yml inventory file and describe the network structure of solution components in that file. The installer uses the single.inventory.yml file to deploy KUMA.

3. Install the solution.

   Install the solution and log in to the web interface using the default credentials.

If necessary, you can move solution components to different servers to continue with a distributed configuration.

Preparing the single.inventory.yml inventory file

KUMA components can be installed, updated, and removed in the directory containing the extracted installer by using the Ansible® tool and the user-created YML *inventory file* containing a list of the hosts of KUMA components and other settings. If you want to install all KUMA components on the same server, you must specify the same host for all components in the inventory file.

To create an inventory file for installation on a single server:

1. Copy the archive with the `kuma-ansible-installer-<version>.tar.gz` installer to the server and extract it using the following command (about 2 GB of disk space is required):

   ```
   sudo tar -xpf kuma-ansible-installer-<version name>.tar.gz
   ```

2. Go to the KUMA installer directory by executing the following command:

   ```
   cd kuma-ansible-installer
   ```

3. Copy the single.inventory.yml.template and create an inventory file named single.inventory.yml:

   ```
   cp single.inventory.yml.template single.inventory.yml
   ```

4. Edit the settings in the single.inventory.yml inventory file.

   If you want predefined services to be created during the installation, set deploy_example_services to true.

   ```
   deploy_example_services: true
   ```

   The predefined services will appear only as a result of the initial installation of KUMA. If you are upgrading the system using the same inventory file, the predefined services are not re-created.

5. Replace all `kuma.example.com` strings in the inventory file with the name of the host on which you want to install KUMA components.

The inventory file is created. Now you can use it to install KUMA on a single server.

We recommend backing up the inventory file that you used to install the solution. You can use it to add components to the system or remove KUMA.

Example inventory file for installation on a single server

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: true
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_ssh_private_key_file: <full path to the file with the SSH key>
    ansible_user: <user from the sudo group>
    ansible_become: true
    ansible_port: 2222
  children:
    kuma_core:
      hosts:
        kuma1.example.com:
    kuma_collector:
      hosts:
        kuma1.example.com
    kuma_correlator:
```

```
    hosts:
      kuma1.example.com
  kuma_storage:
    hosts:
      kuma1.example.com:
        shard: 1
        replica: 1
        keeper: 1
  kuma_metrics:
    hosts:
      kuma1.example.com
```

Installing KUMA on a single server

You can install all KUMA components on a single server using the Ansible tool and the single.inventory.yml inventory file.

To install Kuma on a single server:

1. Download the kuma-ansible-installer-<`build number`>.tar.gz KUMA distribution kit to the server and extract it. The archive is extracted into the kuma-ansibleinstaller directory.

2. Go to the directory with the unpacked installer.

3. Depending on the activation type that you want to use to activate the solution, do one of the following:

   - If you want to activate the solution with a key file, place the key file in the <`installer directory`>/roles/kuma/files/ directory.

     The key file must be named license.key.

     `sudo cp <key file>.key <installer directory>/roles/kuma/files/license.key`

   - If you want to activate the solution with a license code, go to the next step of the instructions.

     > Activation with a license code is available only for KUMA 3.4 and later. For earlier versions of KUMA, you must activate the solution with a key file.

4. Run the following command to start the component installation with your prepared single.inventory.yml inventory file:

   `sudo ./install.sh single.inventory.yml`

   If privilege elevation requires specifying a password, run the following command and specify the password when prompted:

   `sudo ./install.sh single.inventory.yml --ask-become-pass`

5. Accept the terms of the End User License Agreement.

   If you do not accept the terms of the End User License Agreement, the solution will not be installed.

Depending on the type of solution activation, the installer produces one of the following results:

- If you want to activate the solution using a file and have placed the key file in <installer directory>/roles/kuma/files/, running the installer with the single.inventory.yml inventory file installs KUMA Core, all services specified in the inventory file, and OOTB resources. If deploy_example_services=true is set in the inventory file, demo services are installed.

- If you want to activate the solution with a license code or provide a license file later, running the installer with the "single.inventory.yml" inventory file installs only the KUMA Core and the Metrics service.

  To install the services, specify the license code on the command line. Then run the postinstall.sh installer with the "single.inventory.yml" inventory file.

  ```
  sudo ./postinstall.sh single.inventory.yml
  ```

  This creates the specified services. You can select the resources that you want to import from the repository.

6. After the installation is complete, log in to the KUMA web interface and enter the address of the KUMA web interface in the address bar of your browser, then enter your credentials on the login page.

   The address of the KUMA web interface is https://<FQDN of the host where KUMA is installed>:7220.

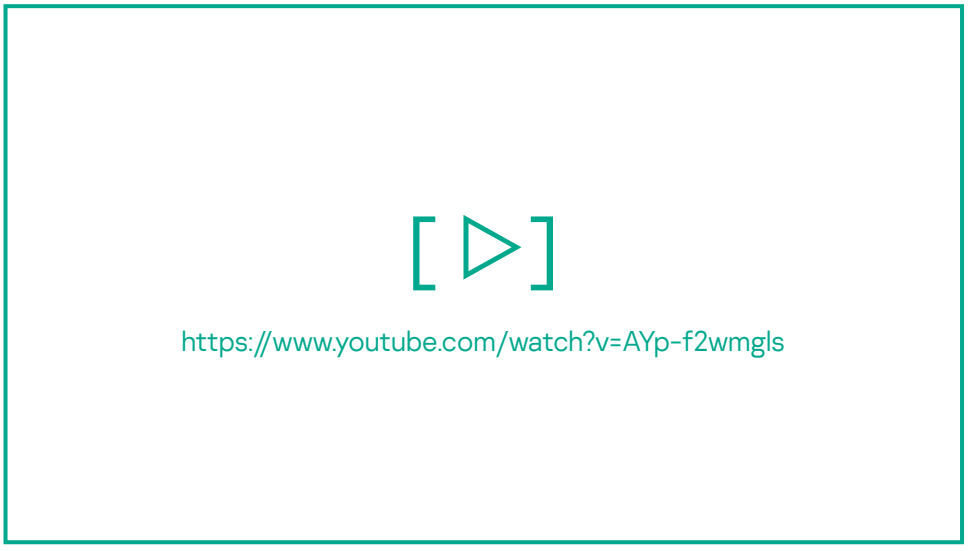
   Default login credentials:
   - login: admin
   - password: mustB3Ch@ng3d!

   After logging in for the first time, change the password of the admin account

All KUMA components are installed and you are logged in to the web interface.

We recommend backing up the inventory file that you used to install the solution. You can use this inventory file to add components to the system or remove KUMA.

You can expand the installation to a distributed installation.



https://www.youtube.com/watch?v=AYp-f2wmgls

Changing the hostname

To change the hostname for a single-server installation:

1. Stop all KUMA services by running the following command:

   ```
   sudo systemctl stop kuma-<service_ID>.service
   ```

2. Create a recovery file by running the following command:

   `sudo touch /opt/kaspersky/kuma/core/<Core_ID>/raft/.reset`

3. Go to the systemd directory with the configuration of the KUMA Core service and change the startup parameters:

   - Specify the new FQDN of the node in the `--raft.node.addr` option.
   - Remove the `--raft.join` option if it is specified.

4. In the systemd configuration files of other services (collector, correlator, storage, event router) change the following settings:

   - In the `--core` option, specify the new name of the node with the KUMA Core.
   - Specify the new computer name in the `--advertise.fqdn` option.

5. Apply the new settings by running the following command:

   `sudo systemctl daemon-reload`

6. Change the hostname on the host:

   `sudo hostnamectl set-hostname <new_hostname>`

7. In the /etc/hosts file, replace the old hostname with the new one.

8. Restart the node by running the following command:

   `sudo reboot`

9. After the node boots, connect to the node under the new hostname.

10. Start the KUMA Core service:

    `sudo systemctl restart kuma-core-00000000-0000-0000-0000-000000000000.service`

11. Verify that the KUMA web interface is accessible under the new hostname.

12. Specify the new hostname in the settings of the KUMA storage.

13. In the KUMA web interface, go to the **Resources and services** → **Active services** section, and in the list of services, select the check box for the services that you need, then in the toolbar or in the menu (⋮), click **Reset certificate**.

    KUMA automatically generates a new certificate for each service for which the certificate has been reset. You do not need to restart running services, the new certificate is applied automatically. A stopped service must be restarted to have the new certificate applied.

14. Delete the directories with service certificates by running the following commands:

    `/opt/kaspersky/kuma/<service_name>/<service_ID>/certificates`

    `/opt/kaspersky/kuma/clickhouse/certificates`

15. Restart all KUMA services:

    `sudo systemctl restart kuma-<service_ID>.service`

The KUMA Core node is now working under the new hostname, and the services are displayed in the **Resources and services** → **Active services** section with a green status.

Distributed installation

The distributed installation of KUMA involves multiple steps:

1. Verifying that the hardware, software, and installation requirements for KUMA are satisfied.
2. Preparing the control machine.

   The control machine is used during the solution installation process to extract and run the installer files.
3. Preparing the target machines.

   The components of the solution are installed on the target machines.
4. Preparing the distributed.inventory.yml inventory file.

   Create an inventory file with a description of the network structure of solution components. The installer uses this inventory file to deploy KUMA.
5. Installing the solution.

   Install the solution and log in to the web interface.
6. Creating services.

   Create the client part of the services in the KUMA web interface and install the server part of the services on the target machines.

   Make sure the KUMA installation is complete before you install KUMA services. We recommend installing services in the following order: storage, collectors, correlators, agents.

   > When deploying several KUMA services on the same host, during installation, you must specify unique ports for each service using the `--api.port <port>` parameter.

If necessary, you can change the KUMA web console certificate to your company's certificate.



https://www.youtube.com/watch?v=-cNRCxNzV2A

Preparing the control machine

To prepare the control machine for installing KUMA:

1. Ensure that hardware, software, and installation requirements of the solution are met.

2. Generate an SSH key for authentication on the SSH servers of the target computers:

   ```
   sudo ssh-keygen -f /root/.ssh/id_rsa -N "" -C kuma-ansible-installer
   ```

   If SSH root access is blocked on the control machine, generate an SSH key for authentication on the SSH servers of the target machines for **a user from the sudo group**:

   If the user does not have sudo rights, add the user to the sudo group:

   ```
   usermod -aG sudo user
   ```

   ```
   ssh-keygen -f /home/<name of the user from the sudo group>/.ssh/id_rsa -N "" -C kuma-ansible-installer
   ```

   As a result, the key is generated and saved in the user's home directory. You should specify the full path to the key in the inventory file in the value of the ansible_ssh_private_key_file parameter so that the key is available during installation.

3. Make sure that the control machine has network access to all the target machines by host name and copy the SSH key to each target machine:

   ```
   sudo ssh-copy-id -i /root/.ssh/id_rsa root@<host name of the target machine>
   ```

   If SSH root access is blocked on the control machine and you want to use the SSH key from the home directory of the sudo group user, make sure that the control machine has network access to all target machines by host name and copy the SSH key to each target machine using the following command:

   ```
   ssh-copy-id -i /home/<name of the user in the sudo group>/.ssh/id_rsa <name of the user in the sudo group>@<host name of the target machine>
   ```

4. Copy the archive with the `kuma-ansible-installer-<version>.tar.gz` installer to the control machine and unpack it using the following command (about 2 GB of disk space is required):

   ```
   sudo tar -xpf kuma-ansible-installer-<version>.tar.gz
   ```

The control machine is prepared for installing KUMA.

Preparing the target machine

To prepare the target machine for the installation of KUMA components:

1. Make sure that hardware, software, and installation requirements are met.

2. Specify the host name. We recommend specifying a FQDN. For example, kuma1.example.com.

   Do not change 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 solution components.

3. Register the target machine in your organization's DNS zone to allow host names to be resolved to IP addresses.

   If your organization does not use a DNS server, you can use the /etc/hosts file for name resolution. The content of the files can be automatically generated for each target machine when installing KUMA.

4. To get the hostname that you must specify when installing KUMA, run the following command and record the result:

   ```
   hostname -f
   ```

   The control machine must be able to access the target machine using this name.

The target machine is ready for the installation of KUMA components.

Preparing the distributed.inventory.yml inventory file

To create the distributed.inventory.yml inventory file:

1. Go to the KUMA installer folder by executing the following command:

   ```
   cd kuma-ansible-installer
   ```

2. Create an inventory file named distributed.inventory.yml by copying distributed.inventory.yml.template:

   ```
   cp distributed.inventory.yml.template distributed.inventory.yml
   ```

3. Edit the settings in the distributed.inventory.yml.

We recommend backing up the inventory file that you used to install the solution. You can use it to add components to the system or remove KUMA.

Example inventory file for distributed installation

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma-core-1.example.com:
          ip: 0.0.0.0
    kuma_collector:
      hosts:
        kuma-collector-1.example.com:
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
          ip: 0.0.0.0
    kuma_storage:
      hosts:
        kuma-storage-cluster1-server1.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 1
```

```
    keeper: 0
  kuma-storage-cluster1-server2.example.com:
    ip: 0.0.0.0
    shard: 1
    replica: 2
    keeper: 0
  kuma-storage-cluster1-server3.example.com:
    ip: 0.0.0.0
    shard: 2
    replica: 1
    keeper: 0
  kuma-storage-cluster1-server4.example.com:
    ip: 0.0.0.0
    shard: 2
    replica: 2
    keeper: 0
  kuma-storage-cluster1-server5.example.com:
    ip: 0.0.0.0
    shard: 0
    replica: 0
    keeper: 1
  kuma-storage-cluster1-server6.example.com:
    ip: 0.0.0.0
    shard: 0
    replica: 0
    keeper: 2
  kuma-storage-cluster1-server7.example.com:
    ip: 0.0.0.0
    shard: 0
    replica: 0
    keeper: 3
```

Installing KUMA in a distributed configuration

KUMA is installed using the Ansible tool and a YML inventory file. The installation is performed from the control machine, and all of the KUMA components are installed on target machines.

To install KUMA:

1. On the control machine, go to the directory containing the extracted installer.

   ```
   cd kuma-ansible-installer
   ```

2. Depending on the activation type that you want to use to activate the solution, do one of the following:

   - To activate the solution using a key file, place the key file in the <installer directory>/roles/kuma/files/ directory.

     The key file must be named license.key.

     ```
     sudo cp <key file>.key <installer directory>/roles/kuma/files/license.key
     ```

   - To activate the solution using an activation code, proceed to the next step of the instructions.

3. From the directory with the unpacked installer, start the installation of components using the prepared distributed.inventory.yml inventory file:

   ```
   sudo ./install.sh distributed.inventory.yml
   ```

If privilege elevation on the target machine requires specifying a password, run the following command and specify the password when prompted:

```
sudo ./install.sh distributed.inventory.yml --ask-become-pass
```

4. If you want to use a non-standard port for the SSH connection during installation, select one of the following options:

   - If the port is the same for a group of hosts, specify it in the inventory file in the 'vars' group.

     Example of a distributed.inventory.yml inventory file with the same non-standard port for a group of storage hosts

     ```
     all:
       vars:
         deploy_to_k8s: false
         need_transfer: false
         generate_etc_hosts: false
         deploy_example_services: false
         no_firewall_actions: false
     kuma:
       vars:
         ansible_connection: ssh
         ansible_user: root
       children:
         kuma_core:
           hosts:
             kuma-core-1.example.com:
               ip: 0.0.0.0
         kuma_collector:
           hosts:
             kuma-collector-1.example.com:
               ip: 0.0.0.0
         kuma_correlator:
           hosts:
             kuma-correlator-1.example.com:
               ip: 0.0.0.0
         kuma_storage:
           vars:
             ansible_port: 2222
           hosts:
             kuma-storage-cluster1-server1.example.com:
               ip: 0.0.0.0
               shard: 1
               replica: 1
               keeper: 0
             kuma-storage-cluster1-server2.example.com:
               ip: 0.0.0.0
               shard: 1
               replica: 2
               keeper: 0
             kuma-storage-cluster1-server3.example.com:
               ip: 0.0.0.0
               shard: 2
               replica: 1
               keeper: 0
             kuma-storage-cluster1-server4.example.com:
               ip: 0.0.0.0
               shard: 2
               replica: 2
     ```

```
        keeper: 0
      kuma-storage-cluster1-server5.example.com:
        ip: 0.0.0.0
        shard: 0
        replica: 0
        keeper: 1
      kuma-storage-cluster1-server6.example.com:
        ip: 0.0.0.0
        shard: 0
        replica: 0
        keeper: 2
      kuma-storage-cluster1-server7.example.com:
        ip: 0.0.0.0
        shard: 0
        replica: 0
        keeper: 3
```

- If a different non-standard port is used for the connection to each host, specify the port for each host in the inventory file, for example:

  Example of a distributed.inventory.yml inventory file with a different non-standard port for each storage host

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_ssh_private_key_file: <full path to the file with the SSH key>
    ansible_user: <user from the sudo group>
    ansible_become: true
  children:
    kuma_core:
      hosts:
        kuma-core-1.example.com:
          raft_node_addr: ""
          ip: 0.0.0.0
    kuma_metrics:
      hosts:
        kuma-metrics.example.com
          ip: 0.0.0.0
    kuma_collector:
      hosts:
        kuma-collector-1.example.com:
          ansible_port: 2222
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
          ansible_port: 2223
          ip: 0.0.0.0
    kuma_storage:
      vars:
        ansible_port: 2224
      hosts:
        kuma-storage-cluster1-server1.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server2.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server3.example.com:
          ip: 0.0.0.0
          shard: 2
```

```
    replica: 1
    keeper: 0
  kuma-storage-cluster1-server4.example.com:
    ip: 0.0.0.0
    shard: 2
    replica: 2
    keeper: 0
  kuma-storage-cluster1-server5.example.com:
    ip: 0.0.0.0
    shard: 0
    replica: 0
    keeper: 1
  kuma-storage-cluster1-server6.example.com:
    ip: 0.0.0.0
    shard: 0
    replica: 0
    keeper: 2
  kuma-storage-cluster1-server7.example.com:
    ip: 0.0.0.0
    shard: 0
    replica: 0
    keeper: 3
```

5. Accept the terms of the End User License Agreement.

   If you do not accept the terms of the End User License Agreement, the solution will not be installed.

   Depending on the type of solution activation, the installer produces one of the following results:

   - If you are activating the solution using a key file and have placed the key file in <installer directory>/roles/kuma/files/, running the installer with the distributed.inventory.yml inventory file installs KUMA Core, all services specified in the inventory file, and OOTB resources.
   - If you are activating the solution with an activation code or plan to upload a key file later, running the installer with the "distributed.inventory.yml" inventory file installs only KUMA Core and the Metrics service.

     To install services, enter the activation code on the command line. Then run the postinstall.sh installer with the distributed.inventory.yml inventory file.

     ```
     sudo ./postinstall.sh distributed.inventory.yml
     ```

     This creates the specified services. You can select the resources that you want to import from the repository.

6. After the installation is complete, log in to the KUMA web interface and enter the address of the KUMA web interface in the address bar of your browser, then enter your credentials on the login page.

   The address of the KUMA web interface is `https://<FQDN of the host where KUMA is installed>:7220`.

   Default login credentials:

   - user name: `admin`
   - password: `mustB3Ch@ng3d!`

   After logging in for the first time, change the password of the admin account

All KUMA components are installed and you are logged in to the web interface.

We recommend backing up the inventory file that you used to install the solution. You can use this inventory file to add components to the system or remove KUMA.

Changing the hostname of microservices

To change the hostname on the node on which the KUMA services are installed:

1. Stop all KUMA services by running the following command:

   ```
   sudo systemctl stop kuma-<service_ID>.service
   ```

2. Go to the services directory, /etc/systemd/system/, and specify the new hostname in the `--advertise.fqdn` option.

3. Apply the changes by running the following command:

   ```
   sudo systemctl daemon-reload
   ```

4. Change the hostname on the node by running the following command:

   ```
   sudo hostnamectl set-hostname <new_hostname>
   ```

5. In the /etc/hosts file, replace the old hostname with the new one.

6. Restart the node by running the following command:

   ```
   sudo reboot
   ```

7. After the node boots, connect to the node under the new hostname.

8. Specify the new hostname in the settings of the KUMA storage.

9. In the KUMA web interface, go to the **Resources and services** → **Active services** section, and in the list of services, select the check box for the services that you need, then in the toolbar or in the menu (⋮), click **Reset certificate**.

   KUMA automatically generates a new certificate for each service for which the certificate has been reset. You do not need to restart running services, the new certificate is applied automatically. A stopped service must be restarted to have the new certificate applied.

10. Delete the directories with service certificates by running the following commands:

    ```
    /opt/kaspersky/kuma/<service_name>/<service_ID>/certificates
    ```

    ```
    /opt/kaspersky/kuma/clickhouse/certificates
    ```

11. Restart all KUMA services:

    ```
    sudo systemctl restart kuma-<service_ID>.service
    ```

The KUMA Core node is now working with the new hostname, and the services are displayed in the **Resources and services** → **Active services** section with a green status.

Distributed installation with KUMA Core in a Kubernetes cluster

The high availability configuration of KUMA involves deploying the KUMA Core on a Kubernetes cluster and using an external TCP traffic balancer.

To create a high availability KUMA installation, use the installer from the kuma-ansible-installer-k8s-<build number>.tar.gz package and prepare the k0s.inventory.yml inventory file by specifying the configuration of the cluster. For a new installation in a high availability configuration, OOTB resources are always imported. You can also perform an installation with deployment of demo services. To do this, set "deploy_example_services: true" in the inventory file.

The KUMA Core can be placed in a Kubernetes cluster in the following ways:

- Install KUMA in a Kubernetes cluster.
- Migrate the Core of an existing KUMA installation to a Kubernetes cluster.

Minimum configuration

Kubernetes has 2 node roles:

- Controllers (control-plane). Nodes with this role manage the cluster, store metadata, and balance the workload.
- Workers (worker). Nodes with this role bear the workload by hosting KUMA processes.

To deploy KUMA in a high availability configuration, you need:

- 3 dedicated controllers
- 2 worker nodes
- 1 TCP balancer

You must not use the balancer as the control machine for running the KUMA installer.

To ensure the adequate performance of the KUMA Core in Kubernetes, you must allocate 3 dedicated nodes that have only the controller role. This will provide high availability for the Kubernetes cluster itself and will ensure that the workload (KUMA processes and other processes) cannot affect the tasks involved in managing the Kubernetes cluster. If you are using virtualization tools, make sure that the nodes are hosted on different physical servers and that these physical servers are not being used as worker nodes.

For a demo installation of KUMA, you may combine the controller and worker roles. However, if you are expanding an installation to a distributed installation, you must reinstall the entire Kubernetes cluster and allocate 3 dedicated nodes with the controller role and at least 2 nodes with the worker role. KUMA cannot be upgraded to later versions if any of the nodes combine the controller and worker roles.

Additional requirements for deploying KUMA Core in Kubernetes

If you plan to protect KUMA's network infrastructure using Kaspersky Endpoint Security for Linux, first install KUMA in the Kubernetes cluster and only then deploy Kaspersky Endpoint Security for Linux. When updating or removing KUMA, you must first stop Kaspersky Endpoint Security for Linux using the following command:

```
systemctl stop kesl
```

When you install KUMA in a high availability configuration, the following requirements must be met:

- General solution installation requirements.
- The hosts that you plan to use for Kubernetes cluster nodes must not use IP addresses from the following Kubernetes ranges:
  - serviceCIDR: 10.96.0.0/12
  - podCIDR: 10.244.0.0/16

  Traffic to proxy servers must excluded for the IP addresses from these ranges.
- Each host must have a unique ID (/etc/machine-id).
- The firewalld or uwf firewall management tool must be installed and enabled on the hosts for adding rules to iptables.
- The nginx load balancer must be installed and configured (for details, please refer to the nginx load balancer documentation). You can install the nginx load balancer using one of the following commands:
  - `sudo yum install nginx` (for Oracle Linux)
  - `sudo apt install nginx-full` (for Astra Linux)
  - `sudo apt install nginx libnginx-mod-stream` (for Ubuntu)
  - `sudo yum install nginx nginx-all-modules` (for RED OS)

  If you want the nginx load balancer to be configured automatically during the KUMA installation, install the nginx load balancer and allow SSH access to it in the same way as for the Kubernetes cluster hosts.

  Example of an automatically created nginx configuration

  The installer creates the `/etc/nginx/kuma_nginx_lb.conf` configuration file. An example of the file contents is shown below. The `upstream` sections are generated dynamically and contain the IP addresses of the Kubernetes cluster controllers (in the example 10.0.0.2-4, in the `upstream kubeAPI_backend`, `upstream konnectivity_backend`, `controllerJoinAPI_backend` sections) and the IP addresses of the worker nodes (in the example 10.0.1.2-3), for which the inventory file contains the `"kaspersky.com/kuma-ingress=true"` value for the `extra_args` variable.

  The `"include /etc/nginx/kuma_nginx_lb.conf;"` line is added to the end of the `/etc/nginx/nginx.conf` file to apply the generated configuration file. With a large number of active services and users, you may need to increase the limit of open files in the `nginx.conf` settings.

  Configuration file example:

```
# Ansible managed
#
# LB KUMA cluster
#

stream {
    server {
        listen          6443;
        proxy_pass      kubeAPI_backend;
    }
    server {
        listen          8132;
        proxy_pass      konnectivity_backend;
```

```
    }
    server {
        listen          9443;
        proxy_pass      controllerJoinAPI_backend;
    }
    server {
        listen          7209;
        proxy_pass      kuma-core-hierarchy_backend;
        proxy_timeout   86400s;
    }
    server {
        listen          7210;
        proxy_pass      kuma-core-services_backend;
        proxy_timeout   86400s;
    }
    server {
        listen          7220;
        proxy_pass      kuma-core-ui_backend;
        proxy_timeout   86400s;
    }
    server {
        listen          7222;
        proxy_pass      kuma-core-cybertrace_backend;
        proxy_timeout   86400s;
    }
    server {
        listen          7223;
        proxy_pass      kuma-core-rest_backend;
        proxy_timeout   86400s;
    }
    upstream kubeAPI_backend {
        server 10.0.0.2:6443;
        server 10.0.0.3:6443;
        server 10.0.0.4:6443;
    }
    upstream konnectivity_backend {
        server 10.0.0.2:8132;
        server 10.0.0.3:8132;
        server 10.0.0.4:8132;
    }
    upstream controllerJoinAPI_backend {
        server 10.0.0.2:9443;
        server 10.0.0.3:9443;
        server 10.0.0.4:9443;
    }
    upstream kuma-core-hierarchy_backend {
        server 10.0.1.2:7209;
        server 10.0.1.3:7209;
    }
    upstream kuma-core-services_backend {
        server 10.0.1.2:7210;
        server 10.0.1.3:7210;
    }
    upstream kuma-core-ui_backend {
        server 10.0.1.2:7220;
        server 10.0.1.3:7220;
    }
    upstream kuma-core-cybertrace_backend {
        server 10.0.1.2:7222;
        server 10.0.1.3:7222;
    }
    upstream kuma-core-rest_backend {
        server 10.0.1.2:7223;
        server 10.0.1.3:7223;
    }
 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.
```

- An access key from the device on which KUMA is installed must be added to the nginx load balancer server.
- On the nginx load balancer server, the SELinux module must be disabled in the operating system.
- The tar, systemctl packages are installed on the hosts.

During KUMA installation, the hosts are automatically checked to see if they meet the following hardware requirements:

- CPU cores (threads): 12 or more
- RAM: 22,528 MB or more
- Free disk space in the /opt partition: 1000 GB or more.
- For an installation from scratch, the /var/lib partition must have at least 32 GB of free space. If the cluster already has been installed on this node, the size of the required free space is reduced by the size of the /var/lib/k0s directory.

If these conditions are not satisfied, the installation is stopped. For a demo installation, you can disable the check of these conditions by setting `low_resources: true` in the inventory file.

Additional requirements when installing on Astra Linux or Ubuntu operating systems.

The following packages must be installed on the machines intended for deploying a Kubernetes cluster:

- open-iscsi
- wireguard
- wireguard-tools

To install the packages, run the following command:

```
sudo apt install open-iscsi wireguard wireguard-tools
```

Additional requirements when installing on the Oracle Linux, RED OS, or Red Hat Enterprise Linux operating systems

The following packages must be installed on the machines intended for deploying the Kubernetes cluster:

- iscsi-initiator-utils
- wireguard-tools

Before installing the packages on Oracle Linux, you must add the EPEL repository as a source of packages using one of the following commands:

- `sudo yum install oracle-epel-release-el8` (for Oracle Linux 8)
- `sudo yum install oracle-epel-release-el9` (for Oracle Linux 9)

To install the packages, run the following command:

```
sudo yum install iscsi-initiator-utils wireguard-tools
```

Installing KUMA in a Kubernetes cluster

The distributed installation of KUMA involves several steps:

1. Verifying that the hardware, software, and installation requirements for KUMA are satisfied.
2. Preparing the control machine.

   The control machine is used during the solution installation process to extract and run the installer files.
3. Preparing the target machines.

   The components of the solution are installed on the target machines.
4. Preparing the k0s.inventory.yml inventory file.

   Create an inventory file with a description of the network structure of solution components. The installer uses this inventory file to deploy KUMA.
5. Installing the solution.

   Install the solution and log in to the web interface.
6. Creating services.

   Create the client part of the services in the KUMA web interface and install the server part of the services on the target machines.

   Make sure the KUMA installation is complete before you install KUMA services. We recommend installing services in the following order: storage, collectors, correlators, agents.

   > When deploying several KUMA services on the same host, during installation, you must specify unique ports for each service using the `--api.port <port>` parameter.

If necessary, you can change the certificate of KUMA web console to use your company's certificate.

Preparing the control machine

To prepare the control machine for installing KUMA:

1. Ensure that hardware, software, and installation requirements of the solution are met.
2. Generate an SSH key for authentication on the SSH servers of the target computers:

   ```
   sudo ssh-keygen -f /root/.ssh/id_rsa -N "" -C kuma-ansible-installer
   ```

   If SSH root access is blocked on the control machine, generate an SSH key for authentication on the SSH servers of the target machines for **a user from the sudo group**:

   If the user does not have sudo rights, add the user to the sudo group:

   ```
   usermod -aG sudo user
   ```

   ```
   ssh-keygen -f /home/<name of the user from the sudo group>/.ssh/id_rsa -N "" -C kuma-ansible-installer
   ```

As a result, the key is generated and saved in the user's home directory. You should specify the full path to the key in the inventory file in the value of the ansible_ssh_private_key_file parameter so that the key is available during installation.

3. Make sure that the control machine has network access to all the target machines by host name and copy the SSH key to each target machine:

   ```
   sudo ssh-copy-id -i /root/.ssh/id_rsa root@<host name of the target machine>
   ```

   If SSH root access is blocked on the control machine and you want to use the SSH key from the home directory of the sudo group user, make sure that the control machine has network access to all target machines by host name and copy the SSH key to each target machine using the following command:

   ```
   ssh-copy-id -i /home/<name of the user in the sudo group>/.ssh/id_rsa <name of the user in the sudo group>@<host name of the target machine>
   ```

4. Copy the installer archive from the `kuma-ansible-installer-k8s-<version number>.tar.gz` package to the control machine and unpack it using the following command:

   ```
   sudo tar -xpf kuma-ansible-installer-k8s-<version number>.tar.gz
   ```

The control machine is prepared for installing KUMA.

Preparing the target machine

To prepare the target machine for the installation of KUMA components:

1. Make sure that hardware, software, and installation requirements are met.

2. Specify the host name. We recommend specifying a FQDN. For example, kuma1.example.com.

   Do not change 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 solution components.

3. Register the target machine in your organization's DNS zone to allow host names to be translated to IP addresses.

   The option of using the /etc/hosts file is not available when the Core is deployed in Kubernetes.

4. To get the hostname that you must specify when installing KUMA, run the following command and record the result:

   ```
   hostname -f
   ```

   The control machine must be able to access the target machine using this name.

The target machine is ready for the installation of KUMA components.

Preparing the k0s.inventory.yml inventory file

To create the k0s.inventory.yml inventory file:

1. Go to the KUMA installer folder by executing the following command:

   ```
   cd kuma-ansible-installer-ha
   ```

2. Copy the k0s.inventory.yml.template file to create the expand.inventory.yml inventory file:

   ```
   cp k0s.inventory.yml.template k0s.inventory.yml
   ```

3. Edit the inventory file settings in k0s.inventory.yml.

Example inventory file for an installation with the KUMA Core in a Kubernetes cluster with 3 controllers, 2 worker nodes, and 1 balancer

```
all:
  vars:
    ansible_connection: ssh
    ansible_ssh_private_key_file: <full path to the SSH key file>
    ansible_user: root
    deploy_to_k8s: true
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
    metrics_disk_request: 150Gi
    core_disk_request: 100Gi
kuma:
  children:
    kuma_collector:
      hosts:
        kuma-collector.example.com:
          ansible_port: 2221
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator.example.com:
          ansible_port: 2222
          ip: 0.0.0.0
    kuma_storage:
      vars:
        ansible_port: 2223
      hosts:
        kuma-storage-cluster1.server1.example.com:
          ip: 0.0.0.0
        kuma-storage-cluster1.server2.example.com:
          ip: 0.0.0.0
        kuma-storage-cluster1.server3.example.com:
          ip: 0.0.0.0
        kuma-storage-cluster1.server4.example.com:
          ip: 0.0.0.0
        kuma-storage-cluster1.server5.example.com:
          ip: 0.0.0.0
        kuma-storage-cluster1.server6.example.com:
          ip: 0.0.0.0
        kuma-storage-cluster1.server7.example.com:
          ip: 0.0.0.0
kuma_k0s:
  children:
    kuma_lb:
      hosts:
        kuma-lb.example.com:
          ip: 0.0.0.0
          kuma_managed_lb: true
    kuma_control_plane_master:
      hosts:
        kuma_cpm.example.com:
          ansible_host: 10.0.1.10
    kuma_control_plane_master_worker:
    kuma_control_plane:
      hosts:
        kuma_cp2.example.com:
          ansible_host: 10.0.1.11
        kuma_cp3.example.com:
          ansible_host: 10.0.1.12
    kuma_control_plane_worker:
    kuma_worker:
      hosts:
        kuma-w1.example.com:
          ansible_host: 10.0.2.11
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
        kuma-w2.example.com:
          ansible_host: 10.0.2.12
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
```

For such a configuration, specify the parameters as follows: need_transfer: false, deploy_example_services: false; in the kuma_storage section, list the servers for the storage cluster. After the installation is complete, you can use the KUMA web interface to assign the shard, replica and keeper roles to the servers specified in the inventory.

Example inventory file for migrating the Core from a distributed installation to a Kubernetes cluster to ensure high availability

```
all:
  vars:
    ansible_connection: ssh
    ansible_user: root
    deploy_to_k8s: true
    need_transfer: true
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
    metrics_disk_request: 150Gi
    core_disk_request: 100Gi
kuma:
  children:
    kuma_collector:
      hosts:
        kuma-collector.example.com:
    kuma_correlator:
      hosts:
        kuma-correlator.example.com:
    kuma_storage:
      hosts:
        kuma-storage-cluster1.server1.example.com
        kuma-storage-cluster1.server2.example.com
        kuma-storage-cluster1.server3.example.com
        kuma-storage-cluster1.server4.example.com
        kuma-storage-cluster1.server5.example.com
        kuma-storage-cluster1.server6.example.com
        kuma-storage-cluster1.server7.example.com
kuma_k0s:
  children:
    kuma_lb:
      hosts:
        kuma-lb.example.com:
          kuma_managed_lb: true
    kuma_control_plane_master:
      hosts:
        kuma_cpm.example.com:
          ansible_host: 10.0.1.10
    kuma_control_plane_master_worker:
    kuma_control_plane:
      hosts:
        kuma_cp2.example.com:
          ansible_host: 10.0.1.11
        kuma_cp3.example.com:
          ansible_host: 10.0.1.12
    kuma_control_plane_worker:
    kuma_worker:
      hosts:
        kuma-w1.example.com:
          ansible_host: 10.0.2.11
```

```
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
        kuma-w2.example.com:
          ansible_host: 10.0.2.12
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
```

The kuma_core, kuma_collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used in the distributed.inventory.yml file when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to subsequent versions, or when a new installation was performed. In the k0s.inventory.yml inventory file, set deploy_to_k8s: true, need_transfer: true, deploy_example_services: false.

We recommend backing up the inventory file that you used to install the solution. You can use it to add components to the system or remove KUMA.

Installation with KUMA Core in a Kubernetes cluster

KUMA is installed using the Ansible tool and the k0s.inventory.yml inventory file. The installation is performed from the control machine, and all of the KUMA components are installed on target machines.

To install KUMA:

1. On the control machine, go to the directory containing the extracted installer:

   ```
   cd kuma-ansible-installer-ha
   ```

2. Depending on the activation type that you want to use to activate the solution, do one of the following:

   - To activate the solution using a key file, place the key file in the <installer directory>/roles/kuma/files/ directory.

     The key file must be named license.key.

     ```
     sudo cp <key file>.key <installer directory>/roles/kuma/files/license.key
     ```

   - To activate the solution using an activation code, proceed to the next step of the instructions.

3. From the folder with the unpacked installer, start the installation of components using the prepared distributed.inventory.yml inventory file:

   ```
   sudo ./install.sh k0s.inventory.yml
   ```

   If privilege elevation on the target machine requires specifying a password, run the following command and specify the password when prompted:

   ```
   sudo ./install.sh k0s.inventory.yml --ask-become-pass
   ```

4. If you want to use a non-standard port for the SSH connection during installation, select one of the following options:

   - If the port is the same for a group of hosts, specify the port in the inventory file in the 'vars' group.

Example of a k0s.inventory.yml inventory file with the same non-standard port for all hosts in the kuma_control_plane section

```
all:
  vars:
    ansible_connection: ssh
    ansible_user: root
    deploy_to_k8s: true
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
    metrics_disk_request: 150Gi
    core_disk_request: 100Gi
kuma:
  children:
    kuma_collector:
      hosts:
        kuma-collector.example.com:
    kuma_correlator:
      hosts:
        kuma-correlator.example.com:
    kuma_storage:
      hosts:
        kuma-storage-cluster1.server1.example.com
        kuma-storage-cluster1.server2.example.com
        kuma-storage-cluster1.server3.example.com
        kuma-storage-cluster1.server4.example.com
        kuma-storage-cluster1.server5.example.com
        kuma-storage-cluster1.server6.example.com
        kuma-storage-cluster1.server7.example.com
kuma_k0s:
  children:
    kuma_lb:
      hosts:
        kuma-lb.example.com:
          kuma_managed_lb: true
    kuma_control_plane_master:
      hosts:
        kuma_cpm.example.com:
          ansible_host: 10.0.1.10
    kuma_control_plane_master_worker:
    kuma_control_plane:
      vars:
        ansible_port: 2222
      hosts:
        kuma_cp2.example.com:
          ansible_host: 10.0.1.11
        kuma_cp3.example.com:
          ansible_host: 10.0.1.12
    kuma_control_plane_worker:
    kuma_worker:
      hosts:
        kuma-w1.example.com:
          ansible_host: 10.0.2.11
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
        kuma-w2.example.com:
          ansible_host: 10.0.2.12
```

```
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
```

- If a different non-standard port is used for the connection to each host, specify the port for each host in the inventory file.

  Example of a k0s.inventory.yml inventory file with a different non-standard port for each hosts in the kuma_control_plane group

```
all:
  vars:
    ansible_connection: ssh
    ansible_user: root
    deploy_to_k8s: true
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
    metrics_disk_request: 150Gi
    core_disk_request: 100Gi
kuma:
  children:
    kuma_collector:
      hosts:
        kuma-collector.example.com:
    kuma_correlator:
      hosts:
        kuma-correlator.example.com:
    kuma_storage:
      hosts:
        kuma-storage-cluster1.server1.example.com
        kuma-storage-cluster1.server2.example.com
        kuma-storage-cluster1.server3.example.com
        kuma-storage-cluster1.server4.example.com
        kuma-storage-cluster1.server5.example.com
        kuma-storage-cluster1.server6.example.com
        kuma-storage-cluster1.server7.example.com
kuma_k0s:
  children:
    kuma_lb:
      hosts:
        kuma-lb.example.com:
          kuma_managed_lb: true
    kuma_control_plane_master:
      hosts:
        kuma_cpm.example.com:
          ansible_host: 10.0.1.10
    kuma_control_plane_master_worker:
    kuma_control_plane:
      hosts:
        kuma_cp2.example.com:
          ansible_port: 2222
          ansible_host: 10.0.1.11
        kuma_cp3.example.com:
          ansible_port: 2223
          ansible_host: 10.0.1.12
    kuma_control_plane_worker:
    kuma_worker:
      hosts:
```

```
        kuma-w1.example.com:
          ansible_host: 10.0.2.11
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
        kuma-w2.example.com:
          ansible_host: 10.0.2.12
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
```

5. Accept the terms of the End User License Agreement.

   If you do not accept the terms of the End User License Agreement, the solution will not be installed.

   Depending on the type of solution activation, the installer produces one of the following results:

   - If you are activating the solution using a key file and have placed the key file in <installer directory>/roles/kuma/files/, running the installer with the k0s.inventory.yml inventory file installs KUMA Core, all services specified in the inventory file, and OOTB resources.

   - If you are activating the solution with an activation code or plan to upload a key file later, running the installer with the "k0s.inventory.yml" inventory file installs only KUMA Core.

     To install services, enter the activation code on the command line. Then run the postinstall.sh installer with the k0s.inventory.yml inventory file.

     ```
     sudo ./postinstall.sh k0s.inventory.yml
     ```

     This creates the specified services. You can select the resources that you want to import from the repository.

6. After the installation is complete, log in to the KUMA web interface and enter the address of the KUMA web interface in the address bar of your browser, then enter your credentials on the login page.

   The address of the KUMA web interface is `https://<FQDN of the nginx load balancer>:7220`.

   Default login credentials:

   - user name: `admin`
   - password: `mustB3Ch@ng3d!`

   After logging in for the first time, change the password of the admin account

All KUMA components are installed and you are logged in to the web interface.

We recommend backing up the inventory file that you used to install the solution. You can use this inventory file to add components to the system or remove KUMA.

Migrating the KUMA Core to a new Kubernetes cluster

To migrate KUMA Core to a new Kubernetes cluster:

1. Prepare the k0s.inventory.yml inventory file.

   The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the KUMA Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed KUMA Core on all hosts where you plan to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

If the component is not detected on the worker nodes, a clean installation of the KUMA Core is performed in the cluster without migrating resources to it. Existing components must be manually rebuilt with the new Core in the KUMA web interface.

Certificates for collectors, correlators and storages will be re-issued from the inventory file for communication with the Core within the cluster. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes systemd services from the host: kuma-core, kuma-vmalert.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.
- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.

- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed is entered into the ConfigMap.

KUMA Core availability under various scenarios

KUMA Core availability in various scenarios:

- **Malfunction or network disconnection of the worker node where the KUMA Core service is deployed.**

  Access to the KUMA web interface is lost. After 6 minutes, Kubernetes initiates migration of the Core bucket to an operational node of the cluster. After the deployment, which takes less than one minute, is completed, the KUMA web interface becomes available again at URLs based on the FQDN of the load balancer. To determine on which of the hosts the Core is running, run the following command in the terminal of one of the controllers:

  ```
  k0s kubectl get pod -n kuma -o wide
  ```

  When the malfunctioning worker node or access to it is restored, the Core bucket is not migrated from its current worker node. The recovered node can participate in the replication of the Core service's disk volume.

- **A worker node that contains a replica of the KUMA Core disk, and which is not hosting the Core service at the moment, fails or loses network connectivity.**

  The KUMA web interface remains available at URLs based on the FQDN of the load balancer. The network storage creates a replica of the running Core disk volume on other running nodes. There is also no disruption of access to KUMA at URLs based on the FQDNs of currently operational nodes.

- **Loss of availability of one or more cluster controllers when quorum is maintained.**

  Worker nodes work normally. Access to KUMA is not disrupted. A failure of cluster controllers extensive enough to break quorum leads to the loss of control over the cluster.

  Correspondence of the number of computers in use to ensure high availability

Number of controllers when installing a cluster	Minimum number of controllers required for the operation of the cluster (quorum)	Admissible number of failed controllers
1	1	0
2	2	0
3	2	1
4	3	1
5	3	2
6	4	2

Number of controllers when installing a cluster	Minimum number of controllers required for the operation of the cluster (quorum)	Admissible number of failed controllers
7	4	3
8	5	3
9	5	4

- **Simultaneous failure of all Kubernetes cluster controllers.**

  The cluster cannot be managed and therefore will have impaired performance.

- **Simultaneous loss of availability of all worker nodes of a cluster with replicas of the Core volume and the Core pod.**

  Access to the KUMA web interface is lost. If all replicas are lost, information will be lost.

Managing Kubernetes and accessing KUMA

When installing KUMA in a high availability configuration, the file named artifacts/k0s-kubeconfig.yml is created in the installer directory. This file contains the details required for connecting to the created Kubernetes cluster. The same file is created on the main controller in the home directory of the user set as ansible_user in the inventory file.

To ensure that the Kubernetes cluster can be monitored and managed, the k0s-kubeconfig.yml file must be saved in a location available for the cluster administrators. Access to the file must be restricted.

Managing a Kubernetes cluster

To monitor and manage the cluster, you can use the k0s application that is installed on all cluster nodes during KUMA deployment. For example, you can use the following command to view the load on worker nodes:

```
k0s kubectl top nodes
```

Access to the KUMA Core

The URL of the KUMA Core is `https://<worker node FQDN>:<worker node port>`. Available ports: 7209, 7210, 7220, 7222, 7223. Port 7220 is used by default to connect to the KUMA Core web interface. Any worker node whose `extra_args` parameter contains the value `kaspersky.com/kuma-ingress=true` can be used as an access point.

It is not possible to log in to the KUMA web interface on multiple worker nodes simultaneously using the same account credentials. Only the most recently established connection remains active.

If you are using an external load balancer in a high availability Kubernetes cluster configuration, the ports of the KUMA Core are accessed via the FQDN of the load balancer.

Time zone in a Kubernetes cluster

The time zone within a Kubernetes cluster is always UTC+0, so this time difference should be taken into account when handling data created by the KUMA Core deployed in a high availability configuration:

- In audit events, the time zone in the `DeviceTimeZone` field is UTC+0.
- In generated reports, the difference between the report generation time and the browser's time will be displayed.
- In the dashboard, the user will find the difference between the time in the widget (the time of the user's browser is displayed) and the time in the exported widget data in the CSV file (the time of the Kubernetes cluster is displayed).

Maintenance of hosts in a Kubernetes cluster

Sometimes you need to restart, install updates, or update the operating system on worker nodes and controllers of the Kubernetes cluster. This article describes a host maintenance procedure that minimizes the downtime of the KUMA Core in a high-availability configuration.

Before taking hosts out for maintenance, you need to create a backup copy of the KUMA core.

Controller maintenance

Cluster controllers must undergo maintenance one at a time. No additional steps are required before performing maintenance of a controller. After maintenance or an upgrade, you must make sure that the controller service has started successfully by checking the status of the service using the following commands:

```
sudo systemctl status k0scontroller
```

```
sudo k0s status
```

After the controller comes back up after maintenance, you can perform maintenance of the next controller. As long as you perform maintenance of controllers one by one, KUMA Core remains available.

Maintenance of working nodes

Worker nodes of the cluster must be taken out for maintenance strictly one at a time.

To perform maintenance on worker nodes:

1. Get the names and current status of worker nodes by running the following command on any controller:

   ```
   sudo k0s kubectl get nodes
   ```

   All worker nodes must have the `ready` status, otherwise the decommissioning of more nodes may lead to unavailability of the KUMA Core.

2. Before upgrading a worker node, you must first prevent pods from being placed on that node. To prevent the placement of pods, run the following command on any controller:

   ```
   sudo k0s kubectl cordon <worker_node_name>
   ```

   After that, the output of the `sudo k0s kubectl get nodes` command shows the node with the `Ready,SchedulingDisabled` status.

3. Get the name of the worker node on which the KUMA Core is running by running the following command on any controller:

   ```
   sudo k0s kubectl get pods -n kuma -o wide
   ```

4. If the KUMA Core is running on the worker node which is undergoing maintenance, restart the KUMA Core deployment to move the KUMA Core to another worker node, and wait until the KUMA Core is successfully started on the other worker node. To restart the deployment of the KUMA Core, run the following command:

   ```
   sudo k0s kubectl rollout restart deployment core-deployment -n kuma
   ```

   > While the KUMA Core is being moved to another worker node, the KUMA Core is unavailable for approximately 10 minutes.

5. Check the status of the KUMA Core by running the following command again:

   ```
   sudo k0s kubectl get pods -n kuma -o wide
   ```

   The pod must have the `Running` status, and the name of the worker node in the `NODE` column must change.

6. Stop the k0sworker service on the node that you are taking out with the following command:

   ```
   sudo k0s stop
   ```

7. Update the operating system.

8. Start the k0s service on the worker node with the following command:

   ```
   sudo k0s start
   ```

9. Make sure that the k0s service has started successfully by running the following commands on the worker node:

   ```
   sudo systemctl status k0sworker
   ```

   ```
   sudo k0s status
   ```

   The k0sworker.service must be in the `active (running)` state.

   The 'k0s status' command should return `Status: Running`.

10. Allow pods to be run on the updated worker node by running the following command on any controller:

    ```
    sudo k0s kubectl uncordon <worker_node_name>
    ```

11. Make sure that the worker node in the cluster is available by running the following command on any controller:

    ```
    sudo k0s kubectl get nodes
    ```

    The updated worker node must have the `ready` status.

12. After returning the working node to the cluster, make sure that all volume replicas are restored and available. To view the status of a volume, run the following command:

    ```
    sudo k0s kubectl get volume -n longhorn-system -o json | jq
    '.items[0].status.robustness'
    ```

The status must be `healthy`. If the status is `degraded`, then one of the replicas is unavailable or is being rebuilt.

13. If you want to monitor the volume rebuilding process, run the following command:

    ```
    sudo k0s kubectl get engine -n longhorn-system -o json | jq
    '.items[0].status.rebuildStatus'
    ```

14. If you want to view the current status of replicas, run the following command:

    ```
    sudo k0s kubectl get replicas -n longhorn-system
    ```

    All replicas must have the `running` status.

Worker node maintenance is complete. If the serviced worker node is ready and the KUMA Core volume has the `healthy` status, you can proceed to perform maintenance on the next worker node.

Maintenance of the traffic balancer

Maintenance of the traffic balancer always results in the KUMA Core becoming temporarily unavailable, both for users and for KUMA services. While the balancer is unavailable, a KUMA Core pot cannot be moved from one worker node to another.

If you are planning a long downtime of the main balancer or substantial upgrades, we recommend the following:

1. Prepare a backup balancer with all necessary updates, duplicate the configuration of nginx or other traffic balancer that you are using from your main balancer to the backup balancer. For Nginx configured by the KUMA installer, the relevant files are /etc/nginx/nginx.conf and /etc/nginx/kuma_nginx_lb.conf. The FQDN of the backup balancer must be the same as the FQDN of the main balancer. You need to make sure that the balancer service has started successfully and that the firewall rules applied to the main balancer are also applicable to the backup load balancer.

2. Switch traffic over to the backup balancer by changing the IP address of the backup balancer to the IP address of the main balancer. On the main balancer, change the IP address to some other IP address to prevent IP address conflict. Make sure that the KUMA Core on the backup balancer is working properly.

   > When you switch over the traffic, current sessions may be terminated. In case of any problems, you can specify the old IP addresses and continue using the main balancer in the previous configuration.

3. Perform maintenance on the main balancer as necessary.

4. Change the IP addresses on the main and backup balancers to their old values.

5. Make sure that the KUMA Core on the updated main balancer is working properly.

The last step may not be necessary if you want to discard the old main balancer, that is, if you are permanently replacing the balancer with an updated host.

The resource requirements of the traffic balancer are minimal, therefore we recommend having a backup clone of the balancer virtual machine on hand to quickly restore the availability of the KUMA Core in case of any problems with the main virtual machine.

Restoring a Kubernetes cluster

This section describes alternatives to using snapshots of virtual machines of cluster nodes as part of a recovery plan.

A prerequisite for recovery is the availability of an up-to-date backup copy of the KUMA Core. We also recommend saving the traffic balancer configuration.

Constituent parts of the Kubernetes cluster and snapshots of their virtual machines

Traffic balancer

After initial configuration, the load balancer is a stateless host. To make a backup copy, you can take a snapshot of its virtual machine at any time after installing the current version of KUMA. Likewise, the virtual machine can be restored from a snapshot at any time, as long as that the KUMA version of the virtual machine snapshot is the same as the current KUMA version. If the versions are different, the traffic balancer configuration file may need to be edited to reflect any changes in the list of served ports.

Limitations:

- Current connections may be broken at the moment when a virtual machine snapshot is created, depending on the type of hypervisor.
- Restoring from a snapshot breaks current connections.
- From the moment when the virtual machine is restored from a snapshot and until new connections are established, the KUMA Core is unavailable.

Cluster controller

Controllers manage the cluster, and the quorum of the controllers must always be maintained during the operation of the cluster.

A controller may be shut down for a short time as long as overall quorum is maintained. Controller virtual machine snapshots can be created one by one. The main limitation is as follows: if the creation of the snapshot causes temporary unavailability of k0s services, depending on the hypervisor, enough time must be allowed between snapshots for the consensus to be restored and for the etcd database of the cluster to be re-synchronized.

Limitations:

- Restoring one or more controllers from a virtual machine snapshot while the cluster is running will lead to undefined behavior.
- A controller virtual machine snapshot may only be used to restore the virtual machine before completely removing the KUMA Core cluster and then reinstalling the KUMA Core from scratch and recovering from a backup copy.

Worker node of the cluster

Worker nodes of the cluster run the KUMA Core and store replicas of the persistent storage volume used by the KUMA Core services. Creating virtual machine snapshots negatively affects the state of the storage volume; from the introduction of errors in the replica with automatic re-creation of the replica to errors in the file system that necessitate completely shutting down the volume and stopping KUMA Core pods. The consequences depend on the type of hypervisor, the operating system of the virtual machine, the state of all replicas and the intensity of disk operations at the time when the snapshot is created.

Limitations:

Restoring a worker node from a virtual machine snapshot while the cluster is running can lead to the cluster becoming inoperable. Even using a snapshot of a temporarily decommissioned working node does not allow the re-introduction of such a node into the cluster after restoring its virtual machine. Snapshots of worker nodes taken just before the installation of the KUMA Core can be useful. Such snapshots may facilitate the preparation of the KUMA Core reinstallation during recovery.

Failure scenarios

Temporary unavailability of individual cluster components

If individual cluster components are temporarily unavailable, for example, due to a temporary server power outage, lack of network connectivity, or a failure that required restarting the server or virtual machine, recovery does not require creating virtual machines from scratch or replacing servers. KUMA Core availability in this case is determined by the set of components that remain in operation. This scenario also covers failures on a server or virtual machine that do not require reinstalling the operating system and that can be quickly remedied by replacing individual hardware components or changing the software configuration.

After restoring the availability of all components, the health of the cluster is restored automatically, however, transient operations take some time to complete, such as the synchronization of volume replicas, during which the cluster remains vulnerable to new failures of other components. Thus, synchronization of replicas of large volumes can take several hours. This recovery time must be taken into account when planning an exercise with the deliberate shutdown of worker nodes.

Complete failure of cluster components while the availability of the KUMA Core is maintained

In this case, the cluster allows using the KUMA Core for some time (until the next component fails), choosing an appropriate time window for recovery, and making an up-to-date backup copy of the KUMA Core.

To restore a cluster:

1. Prepare new virtual machines or servers to replace failed cluster components in accordance with the KUMA installation requirements.

   At this step, you can use snapshots of virtual machines taken before installing KUMA.

2. Review the k0s.inventory.yml inventory file and update it if necessary. If you have many services and you need to minimize installation time, you can leave one host in the `kuma_collector` and `kuma_correlator` sections of the inventory file, and leave one storage cluster in the `kuma_storage` section. If a host from the `kuma_control_plane_master` section has failed, then in inventory file k0s.inventory.yml, you need to swap it with another cluster controller from the `kuma_control_plane` section.

3. Install the current KUMA version using the install.sh script and the prepared k0s.inventory.yml inventory file:

   ```
   sudo ./install.sh k0s.inventory.yml
   ```

4. Make sure that all cluster components are working properly and that high availability is restored:

   a. All k0s services are running:

   ```
   sudo systemctl status <k0sworker/k0scontroller>
   sudo k0s status
   ```

   b. Information about pods and all working nodes is available:

   - To view the status of a volume, run the following command:

     ```
     sudo k0s kubectl get volume -n longhorn-system -o json | jq
     '.items[0].status.robustness'
     ```

     The status must be `healthy`. If the status is `degraded`, then one of the replicas is unavailable or is being restored.

   - To monitor the progress of volume rebuilding, run the following command:

     ```
     sudo k0s kubectl get engine -n longhorn-system -o json | jq
     '.items[0].status.rebuildStatus'
     ```

     If it is working correctly, recovery is not being performed, and the command does not return anything. If the command returns the `rebuilding` status, it means that some replicas are in the process of being rebuilt. We recommend not making any changes to the cluster until the rebuilding is complete.

The cluster is restored.

Complete failure of cluster components with the KUMA Core unavailable

You must have a backup of the KUMA Core on hand.

To restore a cluster, you must first delete the cluster that is in use.

To restore a cluster:

1. Prepare new virtual machines or servers to replace failed cluster components in accordance with the KUMA installation requirements.

   At this step, you can use snapshots of virtual machines taken before installing KUMA.

2. Prepare a separate k0s.inventory.yml inventory file for cluster deletion. In this inventory file, remove all hosts from the `kuma_collector`, `kuma_correlator`, and `kuma_storage` sections to avoid having to uninstall and then reinstall services.

3. Remove the failed cluster:

   a. Run the uninstall.sh script with the k0s.inventory.yml inventory file prepared at step 2:

   ```
   sudo ./uninstall.sh k0s.inventory.yml
   ```

   b. Restart all hosts from the `kuma_worker*` and `kuma_control_plane*` sections of the inventory file.

   c. After starting the hosts from the `kuma_worker*` and `kuma_control_plane*` sections of the inventory file, run the uninstall.sh script again with the k0s.inventory.yml inventory file:

   ```
   sudo ./uninstall.sh k0s.inventory.yml
   ```

4. Prepare the KUMA inventory file for cluster recovery. Use your current inventory file as the basis. If you have many external services and you need to minimize installation time, you can leave one host in the `kuma_collector` and `kuma_correlator` sections of the inventory file, and leave one storage cluster in the `kuma_storage` section. If you do not need to minimize the installation time and restarting of external KUMA services is permissible, then you can use the inventory file without modifications.

5. Install the current KUMA version using the install.sh script and the prepared k0s.inventory.yml inventory file to restore the cluster:

6. Restore the KUMA Core from the backup.

7. Make sure that the KUMA Core and other KUMA services are working properly. To do this, go to the **Resources** → **Active services** section. All services must have the green status.

8. Make sure that all cluster components are working properly and that high availability is restored:

   a. All k0s services are running:

   ```
   sudo systemctl status <k0sworker/k0scontroller>
   sudo k0s status
   ```

   b. Information about pods and all working nodes is available:

   - To view the status of a volume, run the following command:

     ```
     sudo k0s kubectl get volume -n longhorn-system -o json | jq
     '.items[0].status.robustness'
     ```

     The status must be `healthy`. If the status is `degraded`, then one of the replicas is unavailable or is being restored.

   - To monitor the progress of volume rebuilding, run the following command:

     ```
     sudo k0s kubectl get engine -n longhorn-system -o json | jq
     '.items[0].status.rebuildStatus'
     ```

     If it is working correctly, recovery is not being performed, and the command does not return anything. If the command returns the `rebuilding` status, it means that some replicas are in the process of being rebuilt. We recommend not making any changes to the cluster until the rebuilding is complete.

The cluster is restored.

Failure of the traffic balancer

If the traffic balancer fails, the KUMA Core becomes unavailable. However, to restore the balancer, you do not need to delete or change the existing cluster or KUMA services. If you have a snapshot of the balancer virtual machine taken after installing KUMA, you can use this snapshot. If you do not have a virtual machine snapshot or if you want to use a new server to replace the failed one, to restore the cluster, you need to install the balancer and configure it using the previously saved configuration.

After the traffic balancer is recovered, access to the cluster and the KUMA Core is restored.

Distributed installation with KUMA Core in a Raft cluster

A distributed installation of KUMA with multiple KUMA Core services in a cluster helps provide high availability. One of the services becomes the leader; the leader is elected based on the Raft algorithm. Write operations are performed by the leader and broadcast to all servers in the cluster. Read operations are performed by each service from its local SQLite database. Thus, in addition to high availability, creating multiple KUMA Core services also takes care of horizontal scaling. The number of deployed KUMA Core services must be odd.

If necessary, you can:

- **Perform a distributed installation with KUMA Core in a Raft cluster using the install.sh installer and the distributed.inventory.yml inventory file**.

  In the distributed.inventory.yml inventory file, you must manually add the kuma_core_peers group and specify an even number of servers on which you want to install the additional KUMA Core services. Also, in the kuma_core group, you can specify the raft_node_addr parameter in the `<FQDN of a server from the kuma_core group on which the KUMA Core is installed>:<port>` format. This FQDN will be used to listen for signals from other nodes.

  After starting the install.sh installer with the prepared distributed.inventory.yml inventory file, none of the additional KUMA Core services are automatically installed on the servers. The installer only prepares the servers specified in the inventory file for the installation of services. After the installer finishes, only one KUMA Core service is automatically installed on the server specified in the kuma_core group. After the installer finishes, you need to manually create the KUMA Core services in the web interface and install the KUMA Core services with unique IDs one by one on each server specified in the kuma_core_peers group.

- **Expand the cluster and add the necessary number of KUMA Core services using the expand.sh installer and the expand.inventory.yml inventory file**.

  In the expand.inventory.yml inventory file, you must manually add the kuma_core_peers group and specify an even number of servers on which you want to install the additional KUMA Core services.

  After starting the expand.sh installer with the prepared expand.inventory.yml inventory file, none of the additional KUMA Core services are automatically installed on the servers. The installer only prepares the servers specified in the inventory file for the installation of services. After the installer finishes, only one KUMA Core service is automatically installed on the server specified in the kuma_core group. After the installer finishes, you need to manually create the KUMA Core services in the web interface and install the KUMA Core services with unique IDs one by one on each server specified in the kuma_core_peers group.

- **Upgrade a distributed installation of KUMA 3.4 to a distributed installation with KUMA Core in a Raft cluster using the install.sh installer and the distributed.inventory.yml inventory file**.

  In the distributed.inventory.yml inventory file, you must manually add the kuma_core_peers group and specify an even number of servers on which you want to install the additional KUMA Core services. Also, in the kuma_core group, you can specify the raft_node_addr parameter in the `<FQDN of a server from the kuma_core group on which the KUMA Core is installed>:<port>` format. This FQDN will be used to listen for signals from other nodes.

  After starting the installer with the prepared inventory file, none of the additional KUMA Core services are automatically installed on the servers. The installer only prepares the servers specified in the inventory file for the installation of services. After the installer finishes, only one KUMA Core service is automatically installed on the server specified in the kuma_core group. After the installer finishes, you need to manually create the KUMA Core services in the web interface and install the KUMA Core services with unique IDs one by one on each server specified in the kuma_core_peers group.

After installation, all KUMA Core services are automatically added to the Raft cluster.

KUMA Core logs are stored in the /opt/kaspersky/kuma/core/<KUMA Core service ID>/log/core directory.

Preparing the control machine

To prepare the control machine for installing KUMA:

1. Ensure that hardware, software, and installation requirements of the solution are met.

2. Generate an SSH key for authentication on the SSH servers of the target computers:

   ```
   sudo ssh-keygen -f /root/.ssh/id_rsa -N "" -C kuma-ansible-installer
   ```

   If SSH root access is blocked on the control machine, generate an SSH key for authentication on the SSH servers of the target machines for **a user from the sudo group**:

   If the user does not have sudo rights, add the user to the sudo group:

   ```
   usermod -aG sudo user
   ```

   ```
   ssh-keygen -f /home/<name of the user from the sudo group>/.ssh/id_rsa -N "" -C kuma-ansible-installer
   ```

   As a result, the key is generated and saved in the user's home directory. You should specify the full path to the key in the inventory file in the value of the ansible_ssh_private_key_file parameter so that the key is available during installation.

3. Make sure that the control machine has network access to all the target machines by host name and copy the SSH key to each target machine:

   ```
   sudo ssh-copy-id -i /root/.ssh/id_rsa root@<host name of the target machine>
   ```

   If SSH root access is blocked on the control machine and you want to use the SSH key from the home directory of the sudo group user, make sure that the control machine has network access to all target machines by host name and copy the SSH key to each target machine using the following command:

   ```
   ssh-copy-id -i /home/<name of the user in the sudo group>/.ssh/id_rsa <name of the user in the sudo group>@<host name of the target machine>
   ```

4. Copy the `kuma-ansible-installer-<version number>.tar.gz` installer archive to the control machine and unpack it using the following command:

   ```
   sudo tar -xpf kuma-ansible-installer-<version>.tar.gz
   ```

The control machine is prepared for installing KUMA.

Preparing the target machine

To prepare the target machine for the installation of KUMA components:

1. Make sure that hardware, software, and installation requirements are met.

2. Specify the host name. We recommend specifying a FQDN. For example, kuma1.example.com.

   Do not change 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 solution components.

3. Register the target machine in your organization's DNS zone to allow host names to be resolved to IP addresses.

   If your organization does not use a DNS server, you can use the /etc/hosts file for name resolution. The content of the files can be automatically generated for each target machine when installing KUMA.

4. To get the hostname that you must specify when installing KUMA, run the following command and record the result:

   ```
   hostname -f
   ```

   The control machine must be able to access the target machine using this name.

The target machine is ready for the installation of KUMA components.

Preparing the distributed.inventory.yml file for installing KUMA with multiple KUMA Core services on a Raft cluster

KUMA components can be installed, updated, and removed in the directory containing the extracted installer by using the Ansible® tool and the user-created YML *inventory file* containing a list of the hosts of KUMA components and other settings.

To create a distributed.inventory.yml file for installing KUMA with multiple KUMA Core services on a Raft cluster:

1. Go to the KUMA installer directory by executing the following command:

   ```
   cd kuma-ansible-installer
   ```

2. Create an inventory file named distributed.inventory.yml by copying distributed.inventory.yml.template:

   ```
   cp distributed.inventory.yml.template distributed.inventory.yml
   ```

3. Edit the settings in the distributed.inventory.yml.

   Manually add the kuma-core-peers group to the inventory file and specify the servers for installing additional KUMA Core services. Together, the kuma-core and kuma-core-peers groups must list an odd number of servers.

   The kuma-core-00000000-0000-0000-0000-000000000000 service is installed on a server from the kuma_core group (displayed as core-1 in the web interface). On all servers from the kuma_core_peers group, the kuma-core-UID services are sequentially installed and started (displayed as core-<FQDN> in the web interface). These services are connected to the core-1 service. Only one KUMA Core service can be installed on each server.

We recommend backing up the inventory file that you used to install the solution. You can use it to add components to the system or remove KUMA.

Example distributed.inventory.yml file for installing KUMA with multiple KUMA Core services on a Raft cluster

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
```

```
children:
  kuma_core:
    hosts:
      kuma-core-1.example.com:
        ip: 0.0.0.0
        raft_node_addr: ""
  kuma_core_peers:
    hosts:
      kuma-core-2.example.com:
        ip: 0.0.0.0
      kuma-core-3.example.com:
        ip: 0.0.0.0
  kuma_collector:
    hosts:
      kuma-collector-1.example.com:
        ip: 0.0.0.0
  kuma_correlator:
    hosts:
      kuma-correlator-1.example.com:
        ip: 0.0.0.0
  kuma_storage:
    hosts:
      kuma-storage-cluster1-server1.example.com:
        ip: 0.0.0.0
        shard: 1
        replica: 1
        keeper: 0
      kuma-storage-cluster1-server2.example.com:
        ip: 0.0.0.0
        shard: 1
        replica: 2
        keeper: 0
      kuma-storage-cluster1-server3.example.com:
        ip: 0.0.0.0
        shard: 2
        replica: 1
        keeper: 0
      kuma-storage-cluster1-server4.example.com:
        ip: 0.0.0.0
        shard: 2
        replica: 2
        keeper: 0
      kuma-storage-cluster1-server5.example.com:
        ip: 0.0.0.0
        shard: 0
        replica: 0
        keeper: 1
      kuma-storage-cluster1-server6.example.com:
        ip: 0.0.0.0
        shard: 0
        replica: 0
        keeper: 2
      kuma-storage-cluster1-server7.example.com:
        ip: 0.0.0.0
        shard: 0
        replica: 0
        keeper: 3
```

Installing in a high availability configuration in a Raft cluster

These instructions are part of the deployment and initial configuration scenario.

The installation is performed from the control computer, and all of the solution's components are installed on target computers.

Before starting the installation, make sure that you have completed the preparation steps. Also make sure you have a valid license key to activate the solution.

To install Kaspersky SIEM in a distributed configuration with the KUMA Core in the Raft cluster, go to the control computer and do the following:

1. Place the installer archive on the control machine

In your account on the Kaspersky CompanyAccount portal, download the installer archive. Then copy the `kuma-ansible-installer-<KUMA version number>.tar.gz` archive to the control computer and unpack it by using the following command (about 2 GB of disk space is required):

```
tar -xpf kuma-ansible-installer-<KUMA version number>.tar.gz
```

2. Go to the directory with the installer and grant access to files in the directory to the operating system user that is running the KUMA installer.

To do so, run the following sequence of commands:

```
cd kuma-ansible-installer
chown -R <user_name>:<group> kuma-ansible-installer
```

where:

- <user_name> is the operating system user that is running the KUMA installer and needs permissions for the installer directory. This user also needs write permissions for the directory into which the archive will be extracted.
- <group> is the main group of the user.

3 Activate the solution

Depending on the activation method, follow these steps:

- Key file

  Place the key file, `license.key`, in the `<installer directory>/roles/kuma/files` directory:

  ```
  cp <key file>.key <installer directory>/roles/kuma/files/license.key
  ```

- Activation code

  No action required, proceed to the next step of the instructions.

  > The application can be activated using an activation code only in KUMA 3.4 and later versions. For earlier versions of KUMA, you must activate the application with a key file.

4 Prepare an inventory file

1. Create an inventory file named `distributed.inventory.yml` by copying the `distributed.inventory.yml.template` template:

   ```
   cp distributed.inventory.yml.template distributed.inventory.yml
   ```

2. Edit the `distributed.inventory.yml` inventory file settings:

   a. The `distributed.inventory.yml` inventory template provides examples of host names for different KUMA service groups. Before installation, replace these examples with valid FQDNs in the `<host name>.<domain>` format and, if necessary, specify the IPv4 or IPv6 addresses of the infrastructure servers where the corresponding services will be installed.

   b. If you use a non-standard directory for storing SSH keys, specify the full path to the SSH key in the `ansible_ssh_private_key_file` parameter in the inventory file in the `kuma:vars` section to provide access for the installer to target computers.

   c. If you want to use a non-standard port for the SSH connection during installation, select one of the following options:

   - If the port is the same for a group of hosts, specify the port in the `ansible_port` parameter of the inventory file's `vars` section.
   - If a different non-standard port is used to connect to each host, specify the port in the `ansible_port` parameter for each host in the inventory file.

   d. If you want predefined services to be created during the installation, set `deploy_example_services` to `true`.

   The predefined services appear only during the installation of Kaspersky SIEM. If you are upgrading the solution using the same inventory file, the predefined services will not be re-created.

Below is an example of filling out the `distributed.inventory.yml` inventory file. It includes the full path to the SSH key, without deploying of the predefined services. It specifies a common non-standard SSH port for additional KUMA Core hosts, a separate non-standard SSH port for the collector and correlator, and a common non-standard SSH port for the storage host group.

An example of the distributed.inventory.yml inventory file for installation in a highly available configuration in a Kubernetes cluster

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma-core-1.example.com:
          ip: 0.0.0.0
          raft_node_addr: ""
          ansible_port: 2222
    kuma_core_peers:
      vars:
        ansible_port: 2222
      hosts:
        kuma-core-2.example.com:
          ip: 0.0.0.0
        kuma-core-3.example.com:
          ip: 0.0.0.0
    kuma_metrics:
      hosts:
        kuma-metrics.example.com
          ansible_port: 2223
          ip: 0.0.0.0
    kuma_collector:
      hosts:
        kuma-collector-1.example.com:
          ansible_port: 2224
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
          ansible_port: 2225
          ip: 0.0.0.0
    kuma_storage:
      vars:
        ansible_port: 2226
      hosts:
        kuma-storage-cluster1-server1.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server2.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server3.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server4.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server5.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 1
        kuma-storage-cluster1-server6.example.com:
```

```
        ip: 0.0.0.0
        shard: 0
        replica: 0
        keeper: 2
      kuma-storage-cluster1-server7.example.com:
        ip: 0.0.0.0
        shard: 0
        replica: 0
        keeper: 3
```

We recommend backing up the inventory file that you used to install the solution. With its help, you can add components to the solution, update, or remove Kaspersky SIEM.

5 Run the installer

Run the `install.sh` installer with the prepared `distributed.inventory.yml` inventory file:

```
For a user with root rights
```

```
./install.sh distributed.inventory.yml
```

```
To elevate privileges for a user without root rights, a password is required
```

```
./install.sh distributed.inventory.yml --ask-become-pass
```

6 Read the terms of the License Agreement

If you do not accept the terms of the End User License Agreement, the solution will not be installed.

If you want to accept the terms of the License Agreement, when the command prompt is displayed, enter:

```
yes
```

7 Complete the installation

To do this, complete the following steps:

1. To complete the installation, do the following depending on the activation method of the solution:

   Key file

   If you activate the solution using a key file and place the key file in the `<installer directory>/roles/kuma/files/` directory, running the installer with the `distributed.inventory.yml` inventory file, installs the KUMA Core on the server specified in the `kuma_core` section, additional KUMA Cores on the servers specified in the `kuma_core_peers` section, and the Metrics service on the server specified in the `kuma_metrics` section. On the specified hosts listed in the inventory file, the files required for further installation and start of KUMA services are deployed, and pre-configured resources (normalizers, correlation rules, active lists, and others) marked with the **[OOTB]** prefix are available.

Information about the installation progress, successfully completed operations, and encountered errors is logged only in the KUMA Core logs, which are stored in the `/opt/kaspersky/kuma/core/<KUMA Core service ID>/log/core` directory.

After the installer finishes, log in to the web interface.

Activation code

If you activate the solution using an activation code, only KUMA Core on the server from the `kuma_core` section, additional KUMA Cores on the servers from the `kuma_core_peers` section, and Metrics service are installed.

To complete the installation:

1. Log in to the web interface and enter an activation code.
2. Run the `postinstall.sh` installer with the `distributed.inventory.yml` inventory file:

   ```
   ./postinstall.sh distributed.inventory.yml
   ```

   The hosts specified in the inventory file contain the files required for further installation and start of KUMA services. Pre-configured resources (such as normalizers, correlation rules, active lists, and others) marked with the **[OOTB]** prefix are available for import into the repository.

   Information about the installation progress, successfully completed operations, and encountered errors is logged only in the KUMA Core logs, which are stored in the `/opt/kaspersky/kuma/core/<KUMA Core service ID>/log/core` directory.

2. Determine whether you will use the balancer for the Raft cluster and, if necessary, configure the balancer:

   - If the balancer is not used, you have to specify the URLs of the KUMA Core nodes separated by commas during the installation of each service. You can to connect to the KUMA Core servers directly. When switching between nodes, re-authentication is required. After successful login on a new node, the session on the previous node will be terminated.
   - The balancer distributes the load among the cluster nodes and provides a single entry point to the web interface (Single Sign-On, SSO) via AD FS, eliminating the need for repeated authentication when switching between cluster nodes. It also ensures continued access to Kaspersky SIEM if one or more hosts running the KUMA Core service are unavailable or down.

     We recommend using one of the following balancers:

     HAProxy Community Edition

     This article provides a configuration example for HAProxy Community Edition. If necessary, you can change the settings to suit your needs. This configuration was tested with package version 2.4.24-0ubuntu0.22.04.2.

     To configure HAProxy Community Edition:

     1. On the host where you are installing the balancer, start the command line console and install the haproxy package:

        ```
        sudo apt update && sudo apt install haproxy -y
        ```

     2. Go to the /etc/haproxy directory and edit the haproxy.cfg configuration file as shown in the example. The order of sections and parameters in the configuration file is significant and must be the same as in the example.

Example of settings in the haproxy.cfg file if a TLS certificate is not used

```
global
        log /dev/log    local0
        log /dev/log    local1 notice
        chroot /var/lib/haproxy
        stats socket /run/haproxy/admin.sock mode 660 level admin
        stats timeout 30s
        user haproxy
        group haproxy
defaults
        log     global
        mode    tcp
        option  tcplog
        option  dontlognull
        timeout connect 5000
        timeout client  50000
        timeout server  50000
        errorfile 400 /etc/haproxy/errors/400.http
        errorfile 403 /etc/haproxy/errors/403.http
        errorfile 408 /etc/haproxy/errors/408.http
        errorfile 500 /etc/haproxy/errors/500.http
        errorfile 502 /etc/haproxy/errors/502.http
        errorfile 503 /etc/haproxy/errors/503.http
        errorfile 504 /etc/haproxy/errors/504.http
frontend deploy-raft
    mode tcp
    bind *:7220
    default_backend be_d1
backend be_d1
    mode tcp
    balance roundrobin
    server machine2 kuma-inst-02.avp.ru:7220 check
    server machine3 kuma-inst-03.avp.ru:7220 check
    server machine4 kuma-inst-04.avp.ru:7220 check
```

Example of settings in the haproxy.cfg file when using a TLS certificate

```
global #
        log /dev/log    local0
        log /dev/log    local1 notice
        chroot /var/lib/haproxy
        stats socket /run/haproxy/admin.sock mode 660 level admin
        stats timeout 30s
        user haproxy
        group haproxy
# Default SSL material locations
        ca-base /etc/ssl/certs
        crt-base /etc/ssl/private
        ssl-default-bind-ciphers ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-
GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-
AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384
        ssl-default-bind-ciphersuites
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
        ssl-default-bind-options ssl-min-ver TLSv1.3 no-tls-tickets
defaults #
        log     global
        mode    http
        option  httplog
```

```
        option  dontlognull
        timeout connect 5000
        timeout client  50000
        timeout server  50000
        errorfile 400 /etc/haproxy/errors/400.http
        errorfile 403 /etc/haproxy/errors/403.http
        errorfile 408 /etc/haproxy/errors/408.http
        errorfile 500 /etc/haproxy/errors/500.http
        errorfile 502 /etc/haproxy/errors/502.http
        errorfile 503 /etc/haproxy/errors/503.http
        errorfile 504 /etc/haproxy/errors/504.http
frontend http_front
   bind 0.0.0.0:443 ssl crt kuma-inst-01.avp.ru
   default_backend be_d1 #name of the default application server group
backend be_d1 #be_d1 is the name of the application server group; used by the
frontend section for addressing purposes
   balance roundrobin #requests are distributed equally among available
application servers
   cookie SERVER insert indirect nocache #the insert parameter creates a cookie
file upon request; indirect deletes the cookie file upon every incoming request
before forwarding it to the application server, #nocache ensures that there is no
caching between the client and the balancer
   option httpchk #start HTTP check
   http-check connect port 7220 #port 7220 is used for the HTTP check
   http-check send meth GET #HTTP method GET is used for the check
   http-check expect status 200 #a successful HTTP check gets the 200 OK status
from the application server
   server machine2 kuma-inst-02.avp.ru:7220 inter 4s fall 2 rise 4 ssl verify
none #the inter parameter sets the http-check interval to four seconds to exclude
the machine
   server machine3 kuma-inst-03.avp.ru:7220 inter 4s fall 2 rise 4 ssl verify
none #from the rotation, 2 consecutive checks must fail - fall 2, to return, 4
must
   server machine4 kuma-inst-04.avp.ru:7220 inter 4s fall 2 rise 4 ssl verify
none #complete successfully - rise 4. The authenticity of the server TLS
certificate is not verified - ssl  #verify none
```

3. Save the haproxy.cfg configuration file.

4. Check the correctness of the haproxy.cfg configuration file using the following command:

   ```
   haproxy -f /etc/haproxy/haproxy.cfg -c
   ```

   If the check command returns an error, make sure the contents of the haproxy.cfg configuration file matches the example and make changes if necessary. If the check command does not return any errors, proceed to starting the balancer.

5. Start the balancer:

   ```
   sudo systemctl start haproxy
   ```

   If the haproxy service needs to be enabled before starting, run the following sequence of commands:

   ```
   sudo systemctl enable haproxy
   ```

   ```
   sudo systemctl start haproxy
   ```

The balancer is configured.

NGINX Open Source

This article provides an example configuration for NGINX Open Source. If necessary, you can change the settings to suit your needs. This configuration was tested with package version 1.18.0-6ubuntu14.6.

To configure NGINX Open Source:

1. On the host where you are installing the balancer, start the command line console and install the nginx-common package:

   ```
   sudo apt update && sudo apt install nginx-common -y
   ```

2. Go to the /etc/nginx directory and edit the nginx.conf configuration file as shown in the example. The order of sections and parameters in the configuration file is significant and must be the same as in the example.

   Example of settings in the nginx.conf file if a TLS certificate is not used

   ```
   user www-data;
   worker_processes auto;
   pid /run/nginx.pid;
   error_log /var/log/nginx/error.log;
   include /etc/nginx/modules-enabled/*.conf;
   events {
           worker_connections 768;
   }
   http {
           sendfile on;
           tcp_nopush on;
           types_hash_max_size 2048;
           server_tokens off;
           server_name_in_redirect off;
           include /etc/nginx/mime.types;
           default_type application/octet-stream;
           access_log /var/log/nginx/access.log;
           gzip on;
           include /etc/nginx/conf.d/*.conf;
           include /etc/nginx/sites-enabled/*;
   }
   stream {
       include /etc/nginx/stream-conf.d/*.conf;
           upstream backend {
               server kuma-inst-02.avp.ru:7220;
               server kuma-inst-03.avp.ru:7220;
               server kuma-inst-04.avp.ru:7220;
           }
           server {
               listen 7220;
               proxy_pass backend;
           }
   }
   ```

   Example of settings in the nginx.conf file when using a TLS certificate

   ```
   user www-data;
   worker_processes auto;
   pid /run/nginx.pid;
   include /etc/nginx/modules-enabled/*.conf;
   events {
           worker_connections 768;
   }
   http {
   ```

```
        ##
        # Basic Settings
        ##
        sendfile on;
        tcp_nopush on;
        types_hash_max_size 2048;
        server_tokens off;
        server_name_in_redirect off;
        include /etc/nginx/mime.types;
        default_type application/octet-stream;
        ##
        # Logging Settings
        ##
        access_log /var/log/nginx/access.log;
        error_log /var/log/nginx/error.log;
        ##
        # Gzip Settings
        ##
        gzip on; #
        include /etc/nginx/conf.d/*.conf;
        include /etc/nginx/sites-enabled/*;
    server {
        listen              443 ssl;
        server_name         kuma-inst-01.avp.ru;
        ssl_certificate     kuma-inst-01.avp.ru;
        ssl_certificate_key kuma-inst-01.avp.ru.key;
        ssl_protocols       TLSv1.3;
        ssl_ciphers         HIGH:!aNULL:!MD5;
        location / { #path that is accessed to pass requests to the group of
servers from proxy_pass
            proxy_pass https://Kuma-UI; #proxy_pass - the directive for passing
requests to the Kuma-UI server group
        }
    }
    ##
#max_fails—the number of failed solution server availability checks within the
interval.
#fail_timeout - the time after which the computer is excluded from balancing.
    ##

    upstream Kuma-UI {
        hash $request_uri consistent;
        server kuma-inst-02.avp.ru:7220 max_fails=1 fail_timeout=30s;
        server kuma-inst-03.avp.ru:7220 max_fails=1 fail_timeout=30s;
        server kuma-inst-04.avp.ru:7220 max_fails=1 fail_timeout=30s;
    }
}
```

3. Save the nginx.conf configuration file.

4. Check the correctness of the nginx.cfg configuration file using the following command:

   ```
   sudo nginx -f /etc/nginx/nginx.conf -t
   ```

   If the check command returns an error, make sure the contents of the nginx.conf configuration file matches the example and make changes if necessary. If the check command does not return any errors, proceed to starting the balancer.

5. Start the balancer:

```
sudo systemctl start nginx
```

If the nginx service needs to be enabled before starting, run the following sequence of commands:

```
sudo systemctl enable nginx
sudo systemctl start nginx
```

The balancer is configured.

3. If you have configured the use of a load balancer, enable domain authentication via AD FS if necessary:

   a. Specify the FQDN of the balancer in the AD FS settings.

   b. In Kaspersky SIEM, configure domain authentication via AD FS by specifying the FQDN of the balancer in the settings.

4. Set the same time zone on all servers on which the KUMA Core services are deployed. After the installation is complete, you can change the time zone, if necessary; the changes must be applied simultaneously to all nodes of the Raft cluster.

The installation is complete. You can go back to the deployment script and continue from the Post-installation steps.

Configuring the balancer for the Raft cluster

The balancer in the Raft cluster distributes the load among the cluster nodes and provides a single entry point to the KUMA web interface, while also providing the ability to manage KUMA if one or more hosts with the KUMA Core service are unavailable or down.

We recommend using one of the following balancers:

- HAProxy Community Edition
- NGINX Open Source

You can use domain authentication via AD FS to avoid having to authenticate separately on each node of the Raft cluster. To do so:

1. Configure the balancer in the Raft cluster.
2. Specify the FQDN of the balancer in the AD FS settings.
3. In KUMA, configure domain authentication via AD FS by specifying the FQDN of the balancer in the settings.

If you are deploying KUMA for the first time, after configuring the load balancer, return to the end-to-end scenario and continue from the Post-installation steps.

Example configuration of HAProxy Community Edition

This article provides a configuration example for HAProxy Community Edition. If necessary, you can change the settings to suit your needs. This configuration was tested with package version 2.4.24-0ubuntu0.22.04.2.

To configure HAProxy Community Edition:

1. On the host where you are installing the balancer, start the command line console and install the haproxy package:

   ```
   sudo apt update && sudo apt install haproxy -y
   ```

2. Go to the /etc/haproxy directory and edit the haproxy.cfg configuration file as shown in the example. The order of sections and parameters in the configuration file is significant and must be the same as in the example.

   Example of settings in the haproxy.cfg file if a TLS certificate is not used

   ```
   global
           log /dev/log    local0
           log /dev/log    local1 notice
           chroot /var/lib/haproxy
           stats socket /run/haproxy/admin.sock mode 660 level admin
           stats timeout 30s
           user haproxy
           group haproxy
   defaults
           log     global
           mode    tcp
           option  tcplog
           option  dontlognull
           timeout connect 5000
           timeout client  50000
           timeout server  50000
           errorfile 400 /etc/haproxy/errors/400.http
           errorfile 403 /etc/haproxy/errors/403.http
           errorfile 408 /etc/haproxy/errors/408.http
           errorfile 500 /etc/haproxy/errors/500.http
           errorfile 502 /etc/haproxy/errors/502.http
           errorfile 503 /etc/haproxy/errors/503.http
           errorfile 504 /etc/haproxy/errors/504.http
   frontend deploy-raft
       mode tcp
       bind *:7220
       default_backend be_d1
   backend be_d1
       mode tcp
       balance roundrobin
       server machine2 kuma-inst-02.avp.ru:7220 check
       server machine3 kuma-inst-03.avp.ru:7220 check
       server machine4 kuma-inst-04.avp.ru:7220 check
   ```

   Example of settings in the haproxy.cfg file when using a TLS certificate

   ```
   global #
           log /dev/log    local0
           log /dev/log    local1 notice
           chroot /var/lib/haproxy
           stats socket /run/haproxy/admin.sock mode 660 level admin
           stats timeout 30s
           user haproxy
           group haproxy
   # Default SSL material locations
           ca-base /etc/ssl/certs
           crt-base /etc/ssl/private
   ```

```
        ssl-default-bind-ciphers ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-
SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-
SHA256:DHE-RSA-AES256-GCM-SHA384
        ssl-default-bind-ciphersuites
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
        ssl-default-bind-options ssl-min-ver TLSv1.3 no-tls-tickets
defaults #
        log     global
        mode    http
        option  httplog
        option  dontlognull
        timeout connect 5000
        timeout client  50000
        timeout server  50000
        errorfile 400 /etc/haproxy/errors/400.http
        errorfile 403 /etc/haproxy/errors/403.http
        errorfile 408 /etc/haproxy/errors/408.http
        errorfile 500 /etc/haproxy/errors/500.http
        errorfile 502 /etc/haproxy/errors/502.http
        errorfile 503 /etc/haproxy/errors/503.http
        errorfile 504 /etc/haproxy/errors/504.http
frontend http_front
   bind 0.0.0.0:443 ssl crt kuma-inst-01.avp.ru
   default_backend be_d1 #name of the default application server group
backend be_d1 #be_d1 is the name of the application server group; used by the frontend
section for addressing purposes
   balance roundrobin #requests are distributed equally among available application
servers
   cookie SERVER insert indirect nocache #the insert parameter creates a cookie file
upon request; indirect deletes the cookie file upon every incoming request before
forwarding it to the application server, #nocache ensures that there is no caching
between the client and the balancer
   option httpchk #start HTTP check
   http-check connect port 7220 #port 7220 is used for the HTTP check
   http-check send meth GET #HTTP method GET is used for the check
   http-check expect status 200 #a successful HTTP check gets the 200 OK status from
the application server
   server machine2 kuma-inst-02.avp.ru:7220 inter 4s fall 2 rise 4 ssl verify none #the
inter parameter sets the http-check interval to four seconds to exclude the machine
   server machine3 kuma-inst-03.avp.ru:7220 inter 4s fall 2 rise 4 ssl verify none
#from the rotation, 2 consecutive checks must fail - fall 2, to return, 4 must
   server machine4 kuma-inst-04.avp.ru:7220 inter 4s fall 2 rise 4 ssl verify none
#complete successfully - rise 4. The authenticity of the server TLS certificate is not
verified - ssl  #verify none
```

3. Save the haproxy.cfg configuration file.

4. Check the correctness of the haproxy.cfg configuration file using the following command:

   ```
   haproxy -f /etc/haproxy/haproxy.cfg -c
   ```

   If the check command returns an error, make sure the contents of the haproxy.cfg configuration file matches the example and make changes if necessary. If the check command does not return any errors, proceed to starting the balancer.

5. Start the balancer:

   ```
   sudo systemctl start haproxy
   ```

   If the haproxy service needs to be enabled before starting, run the following sequence of commands:

```
sudo systemctl enable haproxy
sudo systemctl start haproxy
```

The balancer is configured.

Example configuration of NGINX Open Source

This article provides an example configuration for NGINX Open Source. If necessary, you can change the settings to suit your needs. This configuration was tested with package version 1.18.0-6ubuntu14.6.

To configure NGINX Open Source:

1. On the host where you are installing the balancer, start the command line console and install the nginx-common package:

   ```
   sudo apt update && sudo apt install nginx-common -y
   ```

2. Go to the /etc/nginx directory and edit the nginx.conf configuration file as shown in the example. The order of sections and parameters in the configuration file is significant and must be the same as in the example.

   Example of settings in the nginx.conf file if a TLS certificate is not used

   ```
   user www-data;
   worker_processes auto;
   pid /run/nginx.pid;
   error_log /var/log/nginx/error.log;
   include /etc/nginx/modules-enabled/*.conf;
   events {
           worker_connections 768;
   }
   http {
           sendfile on;
           tcp_nopush on;
           types_hash_max_size 2048;
           server_tokens off;
           server_name_in_redirect off;
           include /etc/nginx/mime.types;
           default_type application/octet-stream;
           access_log /var/log/nginx/access.log;
           gzip on;
           include /etc/nginx/conf.d/*.conf;
           include /etc/nginx/sites-enabled/*;
   }
   stream {
       include /etc/nginx/stream-conf.d/*.conf;
           upstream backend {
               server kuma-inst-02.avp.ru:7220;
               server kuma-inst-03.avp.ru:7220;
               server kuma-inst-04.avp.ru:7220;
           }
           server {
               listen 7220;
               proxy_pass backend;
           }
   }
   ```

   Example of settings in the nginx.conf file when using a TLS certificate

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;
events {
        worker_connections 768;
}
http {
        ##
        # Basic Settings
        ##
        sendfile on;
        tcp_nopush on;
        types_hash_max_size 2048;
        server_tokens off;
        server_name_in_redirect off;
        include /etc/nginx/mime.types;
        default_type application/octet-stream;
        ##
        # Logging Settings
        ##
        access_log /var/log/nginx/access.log;
        error_log /var/log/nginx/error.log;
        ##
        # Gzip Settings
        ##
        gzip on; #
        include /etc/nginx/conf.d/*.conf;
        include /etc/nginx/sites-enabled/*;
    server {
        listen              443 ssl;
        server_name         kuma-inst-01.avp.ru;
        ssl_certificate     kuma-inst-01.avp.ru;
        ssl_certificate_key kuma-inst-01.avp.ru.key;
        ssl_protocols       TLSv1.3;
        ssl_ciphers         HIGH:!aNULL:!MD5;
        location / { #path that is accessed to pass requests to the group of servers
from proxy_pass
            proxy_pass https://Kuma-UI; #proxy_pass - the directive for passing
requests to the Kuma-UI server group
        }
    }
    ##
#max_fails—the number of failed solution server availability checks within the
interval.
#fail_timeout - the time after which the computer is excluded from balancing.
    ##

    upstream Kuma-UI {
        hash $request_uri consistent;
        server kuma-inst-02.avp.ru:7220 max_fails=1 fail_timeout=30s;
        server kuma-inst-03.avp.ru:7220 max_fails=1 fail_timeout=30s;
        server kuma-inst-04.avp.ru:7220 max_fails=1 fail_timeout=30s;
    }
}
```

3. Save the nginx.conf configuration file.

4. Check the correctness of the nginx.cfg configuration file using the following command:

   ```
   sudo nginx -f /etc/nginx/nginx.conf -t
   ```

   If the check command returns an error, make sure the contents of the nginx.conf configuration file matches the example and make changes if necessary. If the check command does not return any errors, proceed to starting the balancer.

5. Start the balancer:

   ```
   sudo systemctl start nginx
   ```

   If the nginx service needs to be enabled before starting, run the following sequence of commands:

   ```
   sudo systemctl enable nginx
   sudo systemctl start nginx
   ```

The balancer is configured.

Expanding the Raft cluster

Expanding the Raft cluster in KUMA involves the following steps:

1. Preparing servers for installing KUMA Core services.

   The expand.sh installer, which you run on the control machine, prepares the necessary files on the target machines for the subsequent installation of additional KUMA Core services.

2. Creating and installing additional KUMA Core services.

   You need to create additional KUMA Core services in the web interface and install the KUMA Core service on the servers specified in the kuma_core_peers group of the expand.inventory.yml inventory file (one server at a time).

3. Updating the distributed.inventory.yml inventory file.

   Before updating KUMA, make sure that the data in the `distributed.inventory.yml` file matches the current system configuration.

Preparing servers for installing KUMA Core services

To prepare servers for the subsequent installation of additional KUMA Core services:

1. Make sure that you are installing additional KUMA Core services using the same version of the installer that was used to install your current KUMA instance. The full version numbers must be identical.

2. On the control machine, go to the directory containing the extracted installer.

   ```
   cd kuma-ansible-installer
   ```

3. Depending on the activation type that you want to use to activate the solution, do one of the following:

   - If you want to activate the solution with a key file, place the key file in the <`installer directory`>/roles/kuma/files/ directory.

     The key file must be named license.key.

```
sudo cp <key file>.key <installer directory>/roles/kuma/files/license.key
```

- If you want to activate the solution with a license code, go to the next step of the instructions.

4. Create an inventory file named expand.inventory.yml by copying the expand.inventory.yml.template file:

   cp expand.inventory.yml.template expand.inventory.yml

   Edit the settings in the expand.inventory.yml inventory file and specify the servers that you want to add in the 'kuma_core_peers' group. Together, the kuma_core and kuma_core_peers groups must contain an odd number of servers.

   Example of an expand.inventory.yml file for adding servers for additional KUMA Core services

```
kuma:
  vars:
    no_firewall_actions: false
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core_peers:
      hosts:
        kuma-core-4.example.com:
          ip: 0.0.0.0
        kuma-core-5.example.com:
          ip: 0.0.0.0
```

5. On the control computer, run the following command as root from the directory with the unpacked installer:

   ./expand.sh expand.inventory.yml

6. Accept the terms of the End User License Agreement.

   If you do not accept the terms of the End User License Agreement, the solution will not be installed.

   Depending on the type of solution activation, the installer produces one of the following results:

   - If you plan to activate the solution using a key file, and the key file is placed in the `<installer directory>`/roles/kuma/files/ directory, the installer started with the expand.inventory.yml file prepares servers specified in the inventory file for installation of additional KUMA Core services.

   - If you want to activate the solution with a license code or provide a key file later, running the installer with the "expand.inventory.yml" inventory file installs only KUMA Core.

Running this command on each target computer specified in the expand.inventory.yml inventory file creates files for creating and installing additional KUMA Core services.

Creating and installing additional KUMA Core services

KUMA Core services consist of a client part and a server part, therefore creating a service involves two steps:

1. Creating the client part of the KUMA Core service in the web interface.

   a. In the **Resources** → **Active services** section, click **Add** and in the drop-down list, select **Add Core service**.

   b. This opens the **Add Core service** window; in that window, specify the name of the service in the **Core service name** field and click **Add**. The added service appears in the **Services** list.

   c. Select the check box next to the added KUMA Core service and in the toolbar, click ⋮ to open a menu, and in the menu, select **Copy ID**. You will need the ID of the KUMA Core service to install the service on the server.

2. Creating the server part of the KUMA Core service.

   Run the following command on the target computer.

   ```
   sudo /opt/kaspersky/kuma/kuma core --raft.join <FQDN of the host from the kuma_core section where the first KUMA Core service is running>:7210 --id <KUMA Core service ID copied from the web interface> --install
   ```

   Repeat the steps to create the client and server parts of the KUMA Core service for each host in the kuma_core_peers group. Only one KUMA Core service can be installed on a host.

Reflecting the changes in the distributed.inventory.yml inventory file

Add the kuma_core_peers group to the distributed.inventory.yml inventory file and specify the servers for the additional KUMA Core services that you specified in the expand.inventory.yml inventory file so that the inventory file distributed.inventory.yml contains up-to-date information in case you need to update KUMA.

Result

The Raft cluster is expanded. KUMA Core services are added to the Raft cluster and connected to core-1. When installing the resto of the KUMA services, you can specify multiple comma-separated URLs in the installation command to connect to the KUMA Core services in the Raft cluster.

Deleting a Raft cluster node

To delete a Raft cluster node:

1. In the web interface, open the **Resources** → **Active services** section, and in the context menu of the KUMA Core service, select **Copy ID**.

2. In the **Active services** section, select the KUMA Core service installed on the server that you want to remove from the Raft cluster and click **Remove**.

3. Go to the server of the KUMA Core service that you want to remove and stop the service using the following command:

```
sudo systemctl stop kuma-core-<core_ID>.service
```

where:

<core_ID> is the KUMA Core ID copied from the web interface.

4. Delete the working directory.

```
sudo rmdir <directory_path>
```

where:

<directory_path> is the full path to the working directory that you want to delete.

5. Remove the systemd service.

The node is deleted from the Raft cluster.

Restoring a Raft cluster node

If one or more nodes are corrupted, but the cluster is still operational, you can follow these steps to restore the Raft cluster nodes:

1. **Removing a node from the Raft cluster**

   You need to remove the corrupted node from the Raft cluster and then replace it with a new node.

2. **Create a node with the KUMA Core service and add it to the cluster using the --raft.join option**

   You can create a new node using the expand.sh installer and the expand.inventory.yml inventory file, create the KUMA Core service in the web interface, and install the KUMA Core service on the server specified in the `kuma_core_peers` group in the expand.inventory.yml inventory file using the installation command with the `--raft.join` option. After you run the installation command, the new node is added to the Raft cluster.

As a result, the new node gets a complete copy of the state of the leader node. However, temporary files, including artifacts of background tasks that may have been present on the abandoned node, are not restored.

Restoring the Raft cluster

If most of the nodes in the Raft cluster are corrupted and the cluster is inoperable, the following recovery methods are available:

- Restoring the Raft cluster from the current state of a selected node.
- Restoring the Raft cluster from backup.

As a result, you get a new cluster with a single node that has the KUMA Core service installed. If you need high availability, you can restore the KUMA Core services on the old nodes or install KUMA Core services on new nodes and add these nodes to the Raft cluster. You do not need to reset certificates of microservices, such as collectors, correlators, storages, and others, but if you are restoring from backup, you may need to change the value of the `--core` parameter in the systemd file to make the services communicate with the correct KUMA Core if the list of cluster nodes is changed as part of the restoration process.

Restoring the Raft cluster from the current state of a node

To restore the Raft cluster from the current state of a selected node:

1. Prepare the cluster for recovery:

   a. Choose the node to become the initial node of the new cluster. On this node, you need to stop the KUMA Core service, but without deleting the working directory and the systemd file.

   b. Stop the KUMA Core service on all nodes in the cluster:

   ```
   sudo systemctl stop kuma-core-<KUMA_Core_service_ID>.service
   ```

   c. Delete the working directories on all nodes in the cluster except the chosen initial node:

   ```
   sudo rmdir <directory_name>
   ```

   d. On all nodes in the cluster except the initial node, remove the systemd services.

   At this point, all KUMA Core services are stopped and the servers are ready for cluster recovery.

2. Restore the Raft cluster with a single KUMA Core:

   a. On the initial node of the new Raft cluster, create the following file:

   ```
   sudo touch /opt/kaspersky/kuma/core/<service_id>/raft/.reset
   ```

   b. On the initial node of the Raft cluster, remove the `--raft.join` parameter in the systemd file and apply the changes by running the following command:

   ```
   sudo systemctl daemon-reload
   ```

   c. On the initial node of the Raft cluster, start the KUMA Core:

   ```
   sudo systemctl start kuma-core-<KUMA Core service ID>.service
   ```

   The Raft cluster is restored with a single KUMA Core. On the rest of the nodes, the KUMA core services are stopped and the nodes are removed from the cluster.

3. If you need a high-availability Raft cluster, perform the restoration procedure on the rest of the cluster nodes:

   - If you want to use the new servers for KUMA Core services, use the expand.sh installer and the expand.inventory.yml inventory file to deploy the KUMA Core services and add the nodes to the Raft cluster.
   - If you want to use the old nodes, reinstall the KUMA Core services with the `--raft.join` option to add the nodes to the Raft cluster using the following command:

     ```
     sudo /opt/kaspersky/kuma/kuma core --raft.join <FQDN of the initial cluster node>:7210 --id <ID of the KUMA Core service copied from the web interface> --install
     ```

The cluster is restored. Certificates of services, such as collectors, correlators, and storages, do not need to be reset.

Restoring the Raft cluster from backup

The backup does not contain temporary files, including artifacts of background tasks.

To restore the Raft cluster from backup:

1. Prepare the cluster for recovery:

   a. Stop the KUMA Core service on each node in the cluster:

   ```
   sudo systemctl stop kuma-core-<KUMA_Core_service_ID>.service
   ```

   b. On each node where the KUMA Core service is installed, delete the working directory:

   ```
   sudo rmdir <directory_name>
   ```

   c. On each node where the KUMA Core service is installed, remove the systemd service.

   At this point, all KUMA Core services are stopped and the servers are ready for cluster recovery.

2. Restore the Raft cluster with a single KUMA Core:

   a. Select the server that will become the new initial node of the Raft cluster and deploy the new initial node of the cluster without the `--raft.join` option:

   ```
   sudo /opt/kaspersky/kuma/kuma core --raft.node.listen 0.0.0.0:7209 --raft.node.addr "$(hostname -f):7209" --health 0.0.0.0:7208 --internal 0.0.0.0:7210 --private 0.0.0.0:7220 --cybertrace 0.0.0.0:7222 --rest 0.0.0.0:7223 --install
   ```

   b. Restore from backup using the REST API.

   The Raft cluster is restored from backup. The initial node of the KUMA Core cluster is restored.

3. If you need a high-availability Raft cluster, perform the restoration procedure on the rest of the cluster nodes:

   - If you want to use the new servers for KUMA Core services, use the expand.sh installer and the expand.inventory.yml inventory file to deploy the KUMA Core services and add the nodes to the Raft cluster.
   - If you want to use the old nodes, reinstall the KUMA Core services with the `--raft.join` option to add the nodes to the Raft cluster using the following command:

     ```
     sudo /opt/kaspersky/kuma/kuma core --raft.join <FQDN of the initial cluster node>:7210 --id <ID of the KUMA Core service copied from the web interface> --install
     ```

The cluster is restored. You do not need to reset certificates of services, such as collectors, correlators, storages, and others, but you may need to change the value of the `--core` parameter in the systemd file to make the services communicate with the correct KUMA Core if the list of cluster nodes is changed as part of the restoration process.

Changing the hostname of the KUMA Core

To change the hostname on one of the Raft cluster nodes:

1. Remove the two KUMA Core nodes on which you do not need to change the hostname by doing the following:

   a. In the KUMA web interface, in the **Active services** section, select the KUMA Core service installed on the server that you want to remove from the Raft cluster and click **Remove**.

   b. Go to the server of the KUMA Core service that you want to remove and stop the service:

   ```
   sudo systemctl stop kuma-core-<KUMA_Core_service_ID>.service
   ```

   c. Create a backup copy of the working directory.

   d. Delete the working directory:

   ```
   sudo rm -rf /opt/kaspersky/kuma/core/<KUMA_Core_ID>
   ```

   e. Delete the systemd service file:

   ```
   sudo rm /usr/lib/systemd/system/kuma-core-<KUMA_Core_service_ID>.service
   ```

   Repeat the steps to remove the second KUMA Core node.

2. Stop the KUMA Core service on the node on which you want to change the hostname, and disable its autostart by running the following commands:

   ```
   sudo systemctl stop kuma-core-<KUMA_Core_service_ID>.service
   sudo systemctl disable --now kuma-core-<KUMA_Core_ID>
   ```

3. Change the hostname on the host:

   ```
   sudo hostnamectl set-hostname <new_hostname>
   ```

4. In the /etc/hosts file, replace the old hostname with the new one.

5. Restart the node:

   ```
   sudo reboot
   ```

6. After the node boots, connect to the node under the new hostname.

7. Go to the systemd directory with the configuration of the KUMA Core service and change the startup parameters:

   - Specify the new FQDN of the node in the `--raft.node.addr` option.
   - Remove the `--raft.join` option if it is specified.

8. Apply the new settings:

   ```
   sudo systemctl daemon-reload
   ```

9. Create a recovery file:

   ```
   sudo -u kuma touch /opt/kaspersky/kuma/core/<KUMA_Core_ID>/raft/.reset
   ```

10. Configure the KUMA Core autostart:

    ```
    sudo systemctl enable --now kuma-core-<KUMA_Core_ID>
    ```

11. Start the KUMA Core:

    ```
    sudo systemctl restart kuma-core-<KUMA_Core_ID>.service
    ```

12. Follow the steps to add the two nodes that you removed at step 1 to the Raft cluster:

    a. In the **Resources** → **Active services** section, click **Add** and in the drop-down list, select **Add Core service**.

    b. This opens the **Add Core service** window; in that window, specify the name of the service in the **Core service name** field and click **Add**. The added service appears in the **Services** list.

    c. Select the check box next to the added KUMA Core service, open the menu in the toolbar ⋮, and select **Copy ID**. You will need the ID of the KUMA Core service to install the service on the server.

    d. Run the following command on the target computer.

    ```
    sudo /opt/kaspersky/kuma/kuma core --raft.join <FQDN of the host from the kuma_core section where the first KUMA Core service is running>:7210 --id <KUMA Core service ID copied from the web interface> --install
    ```

    Repeat the steps to add the other removed node.

KUMA backup

A full backup allows KUMA Core to be restored after critical failures, update errors, migration between KUMA instances, or data loss.

A backup copy contains all the entities required to return the system to an operational state:

- alerts;
- assets and accounts;
- resources (collectors, normalizers, rules, dictionaries, and other configuration objects);
- parameters of integration with external systems.

To migrate or copy resources, use the resource export and import functionality instead. Resource export allows individual configuration objects to be restored without restoring the entire infrastructure. This reduces recovery time after erroneous changes and decreases the risk of losing current configurations of rules, normalizers, and dictionaries.

To ensure backup relevance, we recommend meeting the following conditions:

1. Create a full backup of KUMA Core at least once every seven days.
2. Export resources every 24 hours.
3. Store several generations of backups and monitor the success of their creation.

The following options for creating a backup copy and restoring from backup are available:

- Backup and restore using the REST API.
- Backup and restore using the web interface.

You can also create backup copies of individual partitions.

Special considerations for KUMA backup

- Data may only be restored from a backup if it is restored to the KUMA of the same version as the backup one.

  If you perform a backup in KUMA 4.0 or later, and then restore the backup to another device with KUMA 4.0 using the REST API (`system/backup → system/restore`), alert events are not restored. To restore alert events in ClickHouse, you need to install a storage with the same storage cluster ID, where in the *kuma.alert_events_local* table, event records can be matched to alerts by IDs. If the storage changes, alert events can be viewed provided that a backup copy is created and then the ClickHouse database is restored.

- Backing up collectors is not necessary, except collectors with an SQL connection. When restoring such collectors, you need to revert the ID to the original value.

Creating a backup of the KUMA Core

You can create a backup copy of KUMA Core in one of the following ways:

- Using the REST API.
- In the KUMA web interface.

Creating a backup copy of KUMA Core in the web interface

To create a backup copy of KUMA Core in the web interface:

1. In the KUMA web interface, in the **Resources** → **Active services** section, select the check box next to the KUMA Core service that has the green status.
2. In the menu (⋮), select **Backup** and click **Create a backup copy**.

   The **Create a backup copy** option is available if one KUMA Core 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 KUMA Core 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.

   The backup starts automatically. The backup file is saved to the **Downloads** folder or the folder that you configured in your browser for downloaded files.

   The backup file name includes the name of the service. If you create a backup copy of the same service multiple times, a copy number in parentheses is appended to the service name. For example, if the file is saved for the first time, the file name is 'core-1'. The file of the next backup copy you create is named 'core-1 (1)'. If download completion notification is enabled in your browser settings, a notification will tell you that the backup file was downloaded successfully or that an error occurred.

   Metrics are not included in the backup copy.

A backup copy of the KUMA Core is created.

Restoring the KUMA Core for a single-server installation

You can restore the KUMA Core from a backup copy only within the same version of KUMA. A version conflict error is returned if attempt to restore KUMA from a backup copy created in a different version of KUMA.

Restoring the KUMA Core without changing the host for core-1

This article describes the procedure for restoring the KUMA Core on the same host on which the KUMA Core was installed.

To restore the KUMA Core from a backup copy, use one of the following methods:

- Using the REST API:

```
curl -k --request POST 'https://<ip_kuma>:7223/api/v3/system/restore' --header 'Authorization: Bearer <token>'
--data-binary '@/backup/backup.tar.gz'
```

- In the web interface:

  1. In the web interface, in the **Resources** → **Active services** section, select the check box next to the KUMA Core service that has the green status.

  2. In the menu (⋮) select **Backup** → **Restore**.

  3. This opens a window; in that window, select the backup file that you want to restore.

  If notifications are configured in your browser, a message is displayed in the upper-right corner, telling you that the **KUMA Core is successfully restored**.

The KUMA Core is restored from the backup, and the web interface login page is displayed.

Restoring the KUMA Core with a change of host for core-1

This article describes how to restore the KUMA Core from a backup copy on a new host.

To restore the KUMA Core from a backup copy:

1. Install KUMA on a single server on a new host using the `install.sh` installer and the `single.inventory.yml` inventory file; set

   ```
   deploy_example_services: false
   ```

   Example single.inventory.yml inventory file

   ```
   all:
     vars:
       deploy_to_k8s: false
       need_transfer: false
       generate_etc_hosts: false
       deploy_example_services: true
       no_firewall_actions: false
   kuma:
     vars:
       ansible_connection: ssh
       ansible_ssh_private_key_file: <full path to the file with the SSH key>
       ansible_user: <user from the sudo group>
       ansible_become: true
       ansible_port: 2222
     children:
       kuma_core:
         hosts:
           kuma1.example.com:
       kuma_collector:
         hosts:
           kuma1.example.com
       kuma_correlator:
         hosts:
           kuma1.example.com
       kuma_storage:
         hosts:
           kuma1.example.com:
             shard: 1
             replica: 1
             keeper: 1
       kuma_metrics:
         hosts:
           kuma1.example.com
   ```

   Only the KUMA Core is installed.

2. Restore the KUMA Core from a backup copy in one of the following ways:

   - Using the REST API:

     ```
     curl -k --request POST 'https://<ip_kuma>:7223/api/v3/system/restore' --header 'Authorization: Bearer
     <token>'  --data-binary '@/backup/backup.tar.gz'
     ```

   - In the web interface:

     a. In the web interface, open the **Resources** → **Active services** section and in the menu (⋮), select **Backup** → **Restore**.

     b. This opens a window; in that window, select the backup file that you want to restore.

     If notifications are configured in your browser, a message is displayed in the upper-right corner, telling you that the **KUMA Core is successfully restored**.

3. Install the Metrics service using the `expand.sh` installer and the `expand.inventory.yml` inventory file. Transfer the data from the `metrics` directory to the host with the restored KUMA Core.

The KUMA Core is restored from the backup, and the web interface login page is displayed.

Restoring the KUMA Core for a distributed installation

You can restore the KUMA Core from a backup copy only within the same version of KUMA. A version conflict error is returned if attempt to restore KUMA from a backup copy created in a different version of KUMA.

Restoring the KUMA Core for a distributed installation without changing the host for core-1

To restore the KUMA Core from a backup copy, use one of the following methods:

- Using the REST API:

  ```
  curl -k --request POST '<FQDN>:7223/api/v3/system/restore' --header 'Authorization: Bearer <token>'  --data-
  binary '@/backup/backup.tar.gz'
  ```

  where:

  - <FQDN> is the fully qualified domain name of the KUMA Core server.
  - <token> is the token for API requests.

- In the web interface:

  a. In the web interface, open the **Resources** → **Active services** section and in the menu (⋮), select **Backup** → **Restore**.

  b. This opens a window; in that window, select the backup file that you want to restore.

  If notifications are configured in your browser, a message is displayed in the upper-right corner, telling you that the **KUMA Core is successfully restored**.

The KUMA Core is restored from the backup, and the web interface login page is displayed.

After the restoration is complete, KUMA Core continues to work with existing services. You do not need to additionally configure services.

Restoring the KUMA Core with a change of host for core-1

These instructions describe how to restore the KUMA Core from a backup copy on a new host. The rest of KUMA services remain on their old hosts.

After restoration, you need to reissue the service certificates and reconnect the services to the KUMA Core on the new host.

To restore the KUMA Core from a backup copy:

1. Prepare the new KUMA instance by installing KUMA on a single server on a new host using the `install.sh` installer and the `single.inventory.yml` inventory file; set

   ```
   deploy_example_services: false
   ```

   Only the KUMA Core will be installed on the host specified in the inventory file.

   The new KUMA Core instance does not contain data from the old infrastructure. The backup is restored in this instance.

2. Restore the KUMA Core from a backup copy in one of the following ways:

   - Using the REST API:

     ```
     curl -k --request POST '<FQDN>:7223/api/v3/system/restore' --header 'Authorization: Bearer <token>'  --
     data-binary '@/backup/backup.tar.gz'
     ```

     where:

     - <FQDN> is the fully qualified domain name of the KUMA Core server.
     - <token> is the token for API requests.

   - In the web interface:

     a. In the web interface, open the **Resources** → **Active services** section and in the menu (⋮), select **Backup** → **Restore**.

     b. This opens a window; in that window, select the backup file that you want to restore.

     After the restoration is complete, the login page for the KUMA web interface is displayed.

     If notifications are configured in your browser, a message is displayed in the upper-right corner, telling you that the **KUMA Core is successfully restored**.

   After restoring the KUMA Core from backup, follow these steps so that services can connect to the KUMA Core on the new host.

3. Reissue the certificates of services. To do this, reset the certificates of all services except the KUMA Core:

   a. In the **Resources** → **Active services** section, select the **Select all** check box, then clear the check box for the KUMA Core service.

   b. In the toolbar or menu (⋮), select **Reset certificate**.

      The certificates are reset and automatically reissued for all services except the KUMA Core. Reissuing certificates for the KUMA Core is not required, since after restoring from a backup, the KUMA Core already contains the necessary data, including information about certificates and trusted services that were restored along with the backup.

4. On the host on which the Metrics service is installed, delete the /opt/kaspersky/kuma/metrics/00000000-0000-0000-0000-000000000000/certificates directory. This is necessary so that the next time the Metrics service starts, it does not use the local data of the previous certificate. You do not need to restart the Metrics service separately. At the next step, the change_url.sh script will reconnect the Metrics service with the new KUMA Core and restart the Metrics service.

5. Reconnect all services to the new KUMA Core and restart all services in one of the following ways:

   - Automatically, using the `change_url.sh` script included in the distribution kit.

     The script reconnects the services installed on the hosts specified in the `distributed.inventory.yml` file en masse and restarts these services. The script processes services that are operational (have a green status) at the time when the script is run.

     a. Update the `distributed.inventory.yml` inventory file by specifying the FQDN of the host to which you moved the KUMA Core in the `kuma_core` section and run the `change_url.sh` script using the following command:

     ```
     ./change_url.sh distributed.inventory.yml
     ```

     The script reconnects all services installed on hosts specified in the `distributed.inventory.yml` inventory file to the new KUMA Core, and also restarts all services.

     Services that were non-operational (had a red status) at the moment when the script was run are not reconnected automatically, even if their status changes to green after that. Reconnect such services manually.

   - Manually using a command. Use this method if you need to reconnect an individual service, for example, a service that was not reconnected by the script.

     ```
     sudo /opt/kaspersky/kuma/kuma collector/correlator/storage/eventRouter/metrics --core https://<FQDN>:<port>
     --pair.fingerprint <CA_cert_ID> --id <service_ID> --api.port <API_port> --install
     ```

     where:

     - <FQDN> is the fully qualified domain name of the KUMA Core server.
     - <port> is the port that KUMA Core uses for internal communications (port 7210 is used by default).

- <CA_cert_ID> is the fingerprint ID of the CA certificate used for authentication when connecting the service.

  The `--pair.fingerprint` argument is optional. When creating a service, KUMA automatically adds this option to the installation command and specifies the fingerprint ID.
  You can install a service without specifying the `--pair.fingerprint` option. In this case, KUMA uses an `insecure` value by default, which skips the validation of the KUMA Core CA certificate when establishing a connection. If set to is `insecure`, the fingerprint does not need to be updated after the CA certificate is reissued.

  To make sure the KUMA Core CA certificate gets validated, we recommend keeping the `--pair.fingerprint` option with the fingerprint ID when installing the service. When reissuing a CA certificate, you need to copy the new certificate fingerprint from the web interface and restart the service with the updated fingerprint ID.

- <service_ID> is the service ID copied from the KUMA web interface.

- <API_port> is the port used for communcation with the component being installed.

After the procedure is completed, all KUMA services are connected to the restored KUMA Core on the new host.

Restoring KUMA in a Kubernetes cluster

Temporary unavailability of individual cluster components

If individual cluster components are temporarily unavailable, for example, due to a temporary server power outage, lack of network connectivity, or a failure that required restarting the server or virtual machine, recovery does not require creating virtual machines from scratch or replacing servers. KUMA Core availability in this case is determined by the set of components that remain in operation. This scenario also covers failures on a server or virtual machine that do not require reinstalling the operating system and that can be quickly remedied by replacing individual hardware components or changing the software configuration.

After restoring the availability of all components, the health of the cluster is restored automatically, however, transient operations take some time to complete, such as the synchronization of volume replicas, during which the cluster remains vulnerable to new failures of other components. Thus, synchronization of replicas of large volumes can take several hours. This recovery time must be taken into account when planning an exercise with the deliberate shutdown of worker nodes.

Complete failure of cluster components while the availability of the KUMA Core is maintained

In this case, the cluster allows using the KUMA Core for some time (until the next component fails), choosing an appropriate time window for recovery, and making an up-to-date backup copy of the KUMA Core.

To restore a cluster:

1. Prepare new virtual machines or servers to replace failed cluster components in accordance with the KUMA installation requirements.

   At this step, you can use snapshots of virtual machines taken before installing KUMA.

2. Review the k0s.inventory.yml inventory file and update it if necessary. If you have many services and you need to minimize installation time, you can leave one host in the `kuma_collector` and `kuma_correlator` sections of the inventory file, and leave one storage cluster in the `kuma_storage` section. If a host from the `kuma_control_plane_master` section has failed, then in inventory file k0s.inventory.yml, you need to swap it with another cluster controller from the `kuma_control_plane` section.

3. Install the current KUMA version using the install.sh script and the prepared k0s.inventory.yml inventory file:

   ```
   ./install.sh k0s.inventory.yml
   ```

   The user that will be used to install the solution must have permissions to the installer archive as well as write permissions for the directory in which the archive is to be unpacked.

4. Restore the KUMA Core from a backup copy in one of the following ways:

   - Using the REST API.
   - In the web interface:

     a. In the KUMA web interface, in the **Resources** → **Active services** section, in the menu (⋮), select **Backup** → **Restore**.

     b. In the file selector window that is displayed, select the backup file that you want to use for recovery.

     If notifications are configured in your browser, a message is displayed in the upper right corner, telling you that the KUMA Core is successfully restored.

     After the restoration, you are taken to the login page of the KUMA web interface.

5. Make sure that all cluster components are working properly and that high availability is restored:

   a. All k0s services are running:

   ```
   sudo systemctl status <k0sworker/k0scontroller>
   sudo k0s status
   ```

   b. Information about pods and all working nodes is available:

   - To view the status of a volume, run the following command:

     ```
     sudo k0s kubectl get volume -n longhorn-system -o json | jq
     '.items[0].status.robustness'
     ```

     The status must be `healthy`. If the status is `degraded`, then one of the replicas is unavailable or is being restored.

   - To monitor the progress of volume rebuilding, run the following command:

     ```
     sudo k0s kubectl get engine -n longhorn-system -o json | jq
     '.items[0].status.rebuildStatus'
     ```

     If it is working correctly, recovery is not being performed, and the command does not return anything. If the command returns the `rebuilding` status, it means that some replicas are in the process of being rebuilt. We recommend not making any changes to the cluster until the rebuilding is complete.

6. Install the Metrics service using the expand.sh installer and the expand.inventory.yml inventory file. Transfer the data from the metrics directory to the host with the restored KUMA Core.

The cluster is restored.

Complete failure of cluster components with the KUMA Core unavailable

You must have a backup of the KUMA Core on hand.

To restore a cluster, you must first delete the cluster that is in use.

To restore a cluster:

1. Prepare new virtual machines or servers to replace failed cluster components in accordance with the KUMA installation requirements.

   At this step, you can use snapshots of virtual machines taken before installing KUMA.

2. Prepare a separate k0s.inventory.yml inventory file for cluster deletion. In this inventory file, remove all hosts from the `kuma_collector`, `kuma_correlator`, and `kuma_storage` sections to avoid having to uninstall and then reinstall services.

3. Remove the failed cluster:

   a. Run the uninstall.sh script with the k0s.inventory.yml inventory file prepared at step 2:

      ```
      ./uninstall.sh k0s.inventory.yml
      ```

   b. Restart all hosts from the `kuma_worker*` and `kuma_control_plane*` sections of the inventory file.

   c. After starting the hosts from the `kuma_worker*` and `kuma_control_plane*` sections of the inventory file, run the uninstall.sh script again with the k0s.inventory.yml inventory file:

      ```
      ./uninstall.sh k0s.inventory.yml
      ```

4. Prepare the KUMA inventory file for cluster recovery. Use your current inventory file as the basis. If you have many external services and you need to minimize installation time, you can leave one host in the `kuma_collector` and `kuma_correlator` sections of the inventory file, and leave one storage cluster in the `kuma_storage` section. If you do not need to minimize the installation time and restarting of external KUMA services is permissible, then you can use the inventory file without modifications.

5. Install the current KUMA version using the install.sh script and the prepared k0s.inventory.yml inventory file to restore the cluster:

6. Restore the KUMA Core from a backup copy in one of the following ways:

   - Using the REST API.

   - In the web interface:

     a. In the KUMA web interface, in the **Resources** → **Active services** section, in the menu (⋮), select **Backup** → **Restore**.

     b. In the file selector window that is displayed, select the backup file that you want to use for recovery.

        If notifications are configured in your browser, a message is displayed in the upper right corner, telling you that the KUMA Core is successfully restored.

        After the restoration, you are taken to the login page of the KUMA web interface.

7. Make sure that the KUMA Core and other KUMA services are working properly. To do this, go to the **Resources** → **Active services** section. All services must have the green status.

8. Make sure that all cluster components are working properly and that high availability is restored:

   a. All k0s services are running:

   ```
   sudo systemctl status <k0sworker/k0scontroller>
   sudo k0s status
   ```

   b. Information about pods and all working nodes is available:

   - To view the status of a volume, run the following command:

     ```
     sudo k0s kubectl get volume -n longhorn-system -o json | jq
     '.items[0].status.robustness'
     ```

     The status must be `healthy`. If the status is `degraded`, then one of the replicas is unavailable or is being restored.

   - To monitor the progress of volume rebuilding, run the following command:

     ```
     sudo k0s kubectl get engine -n longhorn-system -o json | jq
     '.items[0].status.rebuildStatus'
     ```

     If it is working correctly, recovery is not being performed, and the command does not return anything. If the command returns the `rebuilding` status, it means that some replicas are in the process of being rebuilt. We recommend not making any changes to the cluster until the rebuilding is complete.

9. Install the Metrics service using the expand.sh installer and the expand.inventory.yml inventory file. Transfer the data from the metrics directory to the host with the restored KUMA Core.

The cluster is restored.

Restoring KUMA Core in a Raft cluster

Restoring the KUMA Core in a Raft cluster without changing the host for core-1

These instructions describe how to restore the KUMA Core from backup on the host from the `kuma_core` section of the inventory file on which the KUMA Core was originally installed. In this scenario, you need to "dismantle" the cluster and remove the KUMA service from the web interface, restore the KUMA Core from a backup copy on the host from the `kuma_core` section of the inventory file, install the KUMA Core service on additional nodes, and combine these into a cluster for high availability, install the Metrics service, and migrate the existing metrics data to the new host.

You can restore the KUMA Core from a backup copy only within the same version of KUMA. A version conflict error is returned if attempt to restore KUMA from a backup copy created in a different version of KUMA.

To restore the KUMA Core from a backup copy:

1. Prepare the cluster for recovery. To do this, on each host specified in the `kuma_core_peers` section in the inventory file, perform the following actions:

   a. In the KUMA web interface, in **the Resources and services** → **Active services** section, select the check box next to the KUMA Core service, and in the toolbar or in the menu (⋮), select **Copy ID**.

   You will need the ID of the KUMA Core service to stop the service on the host.

   b. In the KUMA web interface, in **the Resources and services** → **Active services** section, select the check box next to the KUMA Core service, and in the toolbar or in the menu (⋮), select **Delete**.

   c. On the Raft cluster host where the KUMA Core service is installed, stop the KUMA Core service by running the following command on the command line:

   `sudo systemctl stop kuma-core-<ID of the KUMA Core service copied from the web interface>.service`

   d. On each Raft cluster host where the KUMA Core service is installed, delete the working directory:

   ```
   sudo rmdir <directory_name>
   ```

   e. On the Raft cluster host where the KUMA Core service is installed, remove the systemd service.

   At this stage, all servers are ready for cluster recovery.

2. Restore the KUMA Core from a backup copy in one of the following ways:

   - Using the post/system/restore REST API method.
   - In the web interface:

     a. In the KUMA web interface, in the **Resources and services** → **Active services** section, in the menu (⋮), select **Backup** → **Restore**.

     b. In the file selector window that is displayed, select the backup file that you want to use for recovery.

     If notifications are configured in your browser, a message is displayed in the upper right corner, telling you that the KUMA Core is successfully restored.

     After the restoration of the KUMA Core, you are taken to the login page of the web interface.

3. If you need a high-availability Raft cluster, perform the restoration procedure on the rest of the cluster nodes:

   - If you want to use the new servers for KUMA Core services, use the expand.sh installer and the expand.inventory.yml inventory file to deploy the KUMA Core services and add the nodes to the Raft cluster.
   - If you want to use the old nodes, reinstall the KUMA Core services with the `--raft.join` option to add the nodes to the Raft cluster using the following command:

     ```
     sudo /opt/kaspersky/kuma/kuma core --raft.join <FQDN of the initial cluster node>:7210 --id <ID of the KUMA Core service copied from the web interface> --install
     ```

4. Install the Metrics service using the expand.sh installer and the expand.inventory.yml inventory file. Transfer the data from the metrics directory to the host with the restored KUMA Core.

The restoration of the KUMA Core from a backup copy is completed.

Restoring the KUMA Core in a Raft cluster with a change of host for core-1

These instructions describe how to restore the KUMA Core from a backup copy on a new host. In this scenario, you need to fully "dismantle" the cluster and remove the KUMA service from the web interface, install the KUMA Core from scratch on a new host, restore the KUMA Core from a backup copy on that host, install the KUMA Core service on additional nodes and combine them into cluster for high availability, reconnect services to the KUMA Core on the new host, install the Metrics service, and migrate the existing metrics data to the new host, and reconfigure the load balancer.

You can restore the KUMA Core from a backup copy only within the same version of KUMA. A version conflict error is returned if attempt to restore KUMA from a backup copy created in a different version of KUMA.

To restore the KUMA Core from a backup copy:

1. Prepare the cluster for recovery. To do this, on all Raft cluster hosts, do the following:

   a. In the KUMA web interface, in **the Resources and services** → **Active services** section, select the check box next to the KUMA Core service, and in the toolbar or in the menu (⋮), select **Copy ID**.

   You will need the ID of the KUMA Core service to stop the service on the host.

   b. In the KUMA web interface, in **the Resources and services** → **Active services** section, select the check box next to the KUMA Core service, and in the toolbar or in the menu (⋮), select **Delete**.

   c. On the Raft cluster host where the KUMA Core service is installed, stop the KUMA Core service by running the following command:

   ```
   sudo systemctl stop kuma-core-<ID of the KUMA Core service copied from the web interface>.service
   ```

   d. On each Raft cluster host where the KUMA Core service is installed, delete the working directory:

   ```
   sudo rmdir <directory_name>
   ```

   e. On the Raft cluster host where the KUMA Core service is installed, remove the systemd service.

   At this point, all KUMA Core services are stopped and the servers are ready for cluster recovery.

2. Perform a single-server installation of KUMA on a new host using the install.sh installer and the single.inventory.yml inventory file; set `deploy_example_services:false`.

   Only the KUMA Core will be installed on the host specified in the inventory file.

3. Restore the KUMA Core from a backup copy in one of the following ways:

   - Using the post/system/restore REST API method.

   - In the web interface:

     a. In the KUMA web interface, in the **Resources** → **Active services** section, in the menu (⋮), select **Backup** → **Restore**.

     b. In the file selector window that is displayed, select the backup file that you want to use for recovery.

     If notifications are configured in your browser, a message is displayed in the upper right corner, telling you that the KUMA Core is successfully restored.

The initial node of the KUMA Core cluster is restored. After the restoration of the initial node, you are taken to the login page of the web interface.

4. If you need a high-availability Raft cluster, perform the restoration procedure on the rest of the cluster nodes:

   - If you want to use the new servers for KUMA Core services, use the expand.sh installer and the expand.inventory.yml inventory file to deploy the KUMA Core services and add the nodes to the Raft cluster.

   - If you want to use the old nodes, reinstall the KUMA Core services with the `--raft.join` option to add the nodes to the Raft cluster using the following command:

     ```
     sudo /opt/kaspersky/kuma/kuma core --raft.join <FQDN of the initial cluster node>:7210 --id <ID of the KUMA Core service copied from the web interface> --install
     ```

5. Reconnect the services to the new KUMA Core using the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma collector/correlator/storage/eventRouter --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 web interface> --api.port <port used for communication with the installed component> --install
   ```

6. Install the Metrics service using the expand.sh installer and the expand.inventory.yml inventory file. Transfer the data from the metrics directory to the host with the restored KUMA Core.

7. Reconfigure the load balancer by specifying the hosts on which the KUMA Core service is installed.

The restoration from a backup copy is completed.

Modifying the configuration of KUMA

The KUMA configuration can be modified in the following ways.

- **Expanding an all-in-one installation to a distributed installation**.

  This article describes how to expand a single-server deployment configuration to a distributed configuration, and provides an example of expanding the configuration.

  The transition to a distributed deployment scheme may be required in the following cases:

  - For load balancing when load exceeds the recommended level and performance decreases.
  - Ensuring high fault tolerance of all components.
  - Requirements for high availability of storage services (a large number of complex search requests).
  - IT infrastructure features, such as isolated network segments with low-bandwidth communication channels.
  - Administrative features. For example, there are requirements to share access between the teams–server administrator, requirements of regulators–local data storage.

To expand a single-server deployment configuration to a distributed configuration:

1. Create a backup copy of KUMA.
2. Prepare the target machines for the expansion of a single-server deployment configuration to a distributed configuration.
3. On the control computer, go to the directory with the extracted KUMA installer by running the following command:

   ```
   cd kuma-ansible-installer
   ```

4. Prepare the `expand.inventory.yml` inventory file:

   a. Create an inventory file named `expand.inventory.yml` by copying the `expand.inventory.yml.template` file:

   ```
   cp expand.inventory.yml.template expand.inventory.yml
   ```

   b. In the `expand.inventory.yml` inventory file, in the `kuma_<component name>` sections, enter the FQDNs of the target computers for installing KUMA services.

5. On the control computer, run the following command as root from the directory with the extracted installer:

   ```
   ./expand.sh expand.inventory.yml
   ```

   This command creates installation files of KUMA services on each target computer specified in `expand.inventory.yml`.

6. Reset the certificates for the services that you want to migrate to the target computers specified in the `expand.inventory.yml` inventory file:

   a. In the KUMA web interface, go to the **Resources and services** → **Active services** section.

   b. In the list of services, select the check boxes for the required services and in the context menu, select **Reset certificate**.

   KUMA automatically generates a new certificate for each service for which the certificate has been reset.

7. On the target machines specified in the `expand.inventory.yml` inventory file, install the server parts of the KUMA services whose server and web parts have already been created for the single-server deployment configuration: storages, collectors, correlators, and others necessary services.
8. Restart the KUMA services whose server parts were installed at the previous step:

   a. In the KUMA web interface, go to the **Resources and services** → **Active services** section.

   b. In the list of services, select the check boxes for the required services and in the context menu, select **Restart**.

9. In the `distributed.inventory.yml` inventory file, specify the FQDNs of the target computers added to the configuration so that the file contains up-to-date information in case of a KUMA upgrade.

Example expansion of a single-server deployment configuration to a distributed configuration

A single-server KUMA configuration is deployed in the organization: `kuma.example.com`. The organization's information security officers came to the conclusion that one server is not enough and two additional storage servers and an additional collector server must be added: `storage-1.example.com`, `storage-2.example.com`, and `remote-collector.example.com`.

To make it so, you need to expand the single-server KUMA configuration to a distributed configuration:

1. Create a backup copy of KUMA.
2. Prepare the target machines (`storage-1.example.com`, `storage-2.example.com`, and `remote-collector.example.com`) for adding new storage servers and collector servers to the KUMA configuration.
3. Prepare the `expand.inventory.yml` inventory file:
   a. Create an inventory file named `expand.inventory.yml` by copying the `expand.inventory.yml.template` file:

      ```
      cp expand.inventory.yml.template expand.inventory.yml
      ```

   b. In the `expand.inventory.yml` `kuma_<component name>` file, enter the FQDNs of your target computers for installing storage services, collectors, and other required servers.

Example expand.inventory.yml file for adding new storage servers and collector servers

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_collector:
      hosts:
        remote-collector.example.com:
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma.example.com:
          ip: 0.0.0.0
    kuma_storage:
      hosts:
        kuma.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 0
          keeper: 1
```

```
storage-1.example.com:
  ip: 0.0.0.0
  shard: 1
  replica: 0
  keeper: 0
storage-2.example.com:
  ip: 0.0.0.0
  shard: 1
  replica: 1
  keeper: 0
```

4. On the `kuma.example.com` control computer, run the following command as root from the directory with the extracted installer:

   ```
   ./expand.sh expand.inventory.yml
   ```

   Running this command on each target computer specified in the `expand.inventory.yml` inventory file creates files for creating and installing KUMA services.

5. Reset the certificates for the storage and collector services that you want to migrate to the target computers specified in the `expand.inventory.yml` inventory file:

   a. In the KUMA web interface, go to the **Resources and services** → **Active services** section.

   b. In the list of services, select the check boxes for the storage and collector services and in the context menu, select **Reset certificate**.

   KUMA automatically generates new certificates for the storage and collector services.

6. On the target machines specified in the `expand.inventory.yml` inventory file, install the server parts of the storage and collector whose server and web parts have already been created for the single-server deployment configuration.

7. Restart the storage and collector services:

   a. In the KUMA web interface, go to the **Resources and services** → **Active services** section.

   b. In the list of services, select the check boxes for the storage and collector services and in the context menu, select **Restart**.

8. If necessary, create destinations with up-to-date information about storages and collectors.

9. In the `distributed.inventory.yml` inventory file, specify the FQDNs of the target computers added to the configuration so that the file contains up-to-date information in case of a KUMA upgrade.

The expansion of a single-server deployment configuration to a distributed configuration is completed.

- **Adding servers for collectors to a distributed installation**.

  This article describes how to add one or more servers to an existing infrastructure to then install collectors on these servers to balance the load. You can use these instructions as an example and adapt them to your requirements.

To add servers to a distributed installation:

1. Make sure that the target machines meet the hardware and software requirements, as well as the requirements for devices with Oracle Linux, Astra Linux, Ubuntu, RED OS, M OS operating systems.

2. On the control computer, go to the directory with the extracted KUMA installer by running the following command:

   ```
   cd kuma-ansible-installer
   ```

3. Create an inventory file named `expand.inventory.yml` by copying the `expand.inventory.yml.template` file:

   ```
   cp expand.inventory.yml.template expand.inventory.yml
   ```

4. Edit the settings in the `expand.inventory.yml` inventory file and specify the servers that you want to add in the `kuma_collector` section.

   Example expand.inventory.yml inventory file for adding collector servers

   ```
   kuma:
     vars:
       ansible_connection: ssh
       ansible_user: root
     children:
       kuma_collector:
         hosts:
           kuma-additional-collector1.example.com:
             ip: 0.0.0.0
           kuma-additional-collector2.example.com:
             ip: 0.0.0.0
       kuma_correlator:
         hosts:
       kuma_storage:
         hosts:
   ```

5. On the control computer, run the following command as root from the directory with the extracted installer:

   ```
   ./expand.sh expand.inventory.yml
   ```

   This command creates files for creating and installing the collector on each target computer specified in the `expand.inventory.yml` inventory file.

6. Create and install the collectors. KUMA collectors are composed of a client part and a server part, therefore you need to follow these two steps:

   a. Create the client part of the collector, which includes a resource set and the collector service.

      To create a resource set for a collector, in the KUMA web interface, under **Resources** → **Collectors**, click **Add collector** and edit the settings. For more details, see Creating a collector.

      At the last step of the configuration wizard, after you click **Create and save**, a resource set for the collector is created and the collector service is automatically created. The command for installing the service on the server is also automatically generated. This command is displayed on the screen. Copy the installation command and proceed to the next step.

b. Create the server part of the collector:

1. On the target computer, run the command you copied at the previous step. The command looks as follows, but all parameters are filled in automatically:

```
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 web interface> --install
```

The collector service is installed on the target computer. You can check the status of the service in the web interface under **Resources** → **Active services**.

2. Run the same command on each target computer specified in the `expand.inventory.yml` inventory file.

7. Specify the added servers in the `distributed.inventory.yml` inventory file so that it has up-to-date information in case of a KUMA update.

Servers are successfully added.

- **Adding servers for correlators to a distributed installation**.

This article describes how to add one or more servers to an existing infrastructure to then install the correlator on these servers to balance the load. You can use these instructions as an example and adapt them to your requirements.

To add servers to a distributed installation:

1. Make sure that the target machines meet the hardware and software requirements, as well as the requirements for devices with Oracle Linux, Astra Linux, Ubuntu, RED OS, M OS operating systems.

2. On the control computer, go to the directory with the extracted KUMA installer by running the following command:

```
cd kuma-ansible-installer
```

3. Create an inventory file named `expand.inventory.yml` by copying the `expand.inventory.yml.template` file:

```
cp expand.inventory.yml.template expand.inventory.yml
```

4. Edit the settings in the `expand.inventory.yml` inventory file and specify the servers that you want to add in the `kuma_correlator` section.

Example expand.inventory.yml inventory file for adding correlator servers

```
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_collector:
    kuma_correlator:
      hosts:
        kuma-additional-correlator1.example.com:
          ip: 0.0.0.0
        kuma-additional-correlator2.example.com:
          ip: 0.0.0.0
```

```
    kuma_storage:
      hosts:
```

5. On the control computer, run the following command as root from the directory with the extracted installer:

   ```
   ./expand.sh expand.inventory.yml
   ```

   This command creates files for creating and installing the correlator on each target computer specified in the `expand.inventory.yml` inventory file.

6. Create and install the correlators. KUMA correlators are composed of a client part and a server part, therefore you need to follow these two steps:

   a. Create the client part of the correlator, which includes a resource set and the correlator service.

      To create a resource set for a correlator, in the KUMA web interface, under **Resources** → **Correlators**, click **Add correlator** and edit the settings. For more details, see Creating a correlator.

      At the last step of the configuration wizard, after you click **Create and save**, a resource set for the correlator is created and the correlator service is automatically created. The command for installing the service on the server is also automatically generated. This command is displayed on the screen. Copy the installation command and proceed to the next step.

   b. Create the server part of the correlator:

      1. On the target computer, run the command you copied at the previous step. The command looks as follows, but all parameter values are assigned automatically:

         ```
         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 web interface> --install
         ```

         The correlator service is installed on the target computer. You can check the status of the service in the web interface under **Resources** → **Active services**.

      2. Run the same command on each target computer specified in the `expand.inventory.yml` inventory file.

7. Specify the added servers in the `distributed.inventory.yml` inventory file so that it has up-to-date information in case of a KUMA update.

Servers are successfully added.

- **Adding servers to an existing storage cluster**.

This article describes how to add multiple servers to an existing storage cluster. You can use these instructions as an example and adapt them to your requirements.

To add servers to an existing storage cluster:

1. Make sure that the target machines meet the hardware and software requirements, as well as the requirements for devices with Oracle Linux, Astra Linux, Ubuntu, RED OS, M OS operating systems.

2. On the control computer, go to the directory with the extracted KUMA installer by running the following command:

   ```
   cd kuma-ansible-installer
   ```

3. Create an inventory file named `expand.inventory.yml` by copying the `expand.inventory.yml.template` file:

   ```
   cp expand.inventory.yml.template expand.inventory.yml
   ```

4. Edit the settings in the `expand.inventory.yml` inventory file and specify the servers that you want to add in the `storage` section. In the example, the `storage` section specifies servers for installing two shards, each of which contains two replicas. In the `expand.inventory.yml` inventory file, specify only the FQDN. You will assign the roles of shards and replicas in the KUMA web interface at later steps of the instructions. You can adapt this example file to your needs.

   Example expand.inventory.yml inventory file for adding servers to an existing storage cluster

   ```
   kuma:
     vars:
       ansible_connection: ssh
       ansible_user: root
     children:
       kuma_collector:
       kuma_correlator:
       kuma_storage:
         hosts:
           kuma-storage-cluster1-server8.example.com:
             ip: 0.0.0.0
           kuma-storage-cluster1-server9.example.com:
             ip: 0.0.0.0
           kuma-storage-cluster1-server10.example.com:
             ip: 0.0.0.0
           kuma-storage-cluster1-server11.example.com:
    ip: 0.0.0.0
   ```

5. On the control computer, run the following command as root from the directory with the extracted installer:

   ```
   ./expand.sh expand.inventory.yml
   ```

   Running this command on each target computer specified in the `expand.inventory.yml` inventory file creates files for creating and installing the storage.

6. You do not need to create a separate storage because you are adding servers to an existing storage cluster. Edit the storage settings of the existing cluster:

   a. In the KUMA web interface, in the **Resources** → **Storages** section, select an existing storage and open it for editing.

   b. In the **ClickHouse cluster nodes** section, click **Add nodes** and specify roles in the fields for the new node. The example describes how to specify IDs to add two shards, containing two replicas each, to an existing cluster. You can adapt this example according to your needs.

   Example of adding nodes

   ```
   ClickHouse cluster nodes
   <existing nodes>
   FQDN: kuma-storage-cluster1server8.example.com
   Shard ID: 1
   Replica ID: 1
   Keeper ID: 0
   FQDN: kuma-storage-cluster1server9.example.com
   Shard ID: 1
   Replica ID: 2
   Keeper ID: 0
   FQDN: kuma-storage-cluster1server9.example.com
   Shard ID: 2
   Replica ID: 1
   Keeper ID: 0
   FQDN: kuma-storage-cluster1server10.example.com
   Shard ID: 2
   Replica ID: 2
   Keeper ID: 0
   ```

   c. Save the storage settings.

   Now you can create storage services for each ClickHouse cluster node.

7. Create a storage service:

   a. In the KUMA web interface, under **Resources** → **Active services**, click **Add service**.

   b. This opens the **Choose a service** window; in that window, select the storage you edited at the previous step and click **Create service**. Do the same for each ClickHouse storage node you are adding.

   The number of created services must be the same as the number of nodes being added to the ClickHouse cluster, for example, 4 services for 4 nodes. The created storage services are displayed in the KUMA web interface in the **Resources** → **Active services** section.

8. Install storage services on each server by using the service ID:

   a. In the KUMA web interface, in the **Resources** → **Active services** section, select the storage service that you need and click **Copy ID**.

   The service ID is copied to the clipboard; you will need it for running the service installation command.

b. Run the following command on the target computer:

```
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 web interface>
--install
```

The storage service is installed on the target computer. You can check the status of the service in the web interface under **Resources** → **Active services**.

c. Run the storage service installation command on each target computer listed in the 'storage' section of the `expand.inventory.yml` inventory file, one computer at a time. On each computer, the unique service ID within the cluster must be specified in the installation command.

9. Apply the changes to the running cluster. To do so, in the KUMA web interface, under **Resources** → **Active services**, select the check boxes next to all storage services in the cluster that you are expanding and click **Update configuration**. Changes are applied without stopping services.

10. Specify the added servers in the `distributed.inventory.yml` inventory file so that it has up-to-date information in case of a KUMA update.

Servers are successfully added to a storage cluster.

- **Adding another storage cluster**.

The following instructions describe how to add an extra storage cluster to an existing infrastructure. You can use these instructions as an example and adapt them to suit your needs.

To add a storage cluster:

1. Make sure that the target machines meet the hardware and software requirements, as well as the requirements for devices with Oracle Linux, Astra Linux, Ubuntu, RED OS, M OS operating systems.

2. On the control computer, go to the directory with the extracted KUMA installer by running the following command:

```
cd kuma-ansible-installer
```

3. Create an inventory file named `expand.inventory.yml` by copying the `expand.inventory.yml.template` file:

```
cp expand.inventory.yml.template expand.inventory.yml
```

4. Edit the settings in the `expand.inventory.yml` inventory file and specify the servers that you want to add in the `storage` section. In the following example, the `storage` section specifies servers for installing 3 dedicated keepers and 2 shards, each of which contains 2 replicas. In the `expand.inventory.yml` inventory file, specify only the FQDN. You will assign the roles of keepers, shards, and replicas in the KUMA web interface at later steps of the instructions. You can adapt this example to suit your needs.

Example expand.inventory.yml inventory file for adding a storage cluster

```
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_collector:
    kuma_correlator:
```

```
kuma_storage:
  hosts:
    kuma-storage-cluster2-server1.example.com:
      ip: 0.0.0.0
    kuma-storage-cluster2-server2.example.com:
      ip: 0.0.0.0
    kuma-storage-cluster2-server3.example.com:
      ip: 0.0.0.0
    kuma-storage-cluster2-server4.example.com:
      ip: 0.0.0.0
    kuma-storage-cluster2-server5.example.com:
      ip: 0.0.0.0
    kuma-storage-cluster2-server6.example.com:
      ip: 0.0.0.0
    kuma-storage-cluster2-server7.example.com:
      ip: 0.0.0.0
```

5. On the control computer, run the following command as root from the directory with the extracted installer:

```
./expand.sh expand.inventory.yml
```

Running this command on each target computer specified in the `expand.inventory.yml` inventory file creates files for creating and installing the storage.

6. Create and install the storage. For each storage cluster, you must create a separate storage, that is, 3 storages for 3 storage clusters. A storage is composed of a client part and a server part, therefore you need to follow these two steps:

   a. Create the client part of the storage, which includes a resource set and the storage service:

      1. To create a resource set for a storage, in the KUMA web interface, under **Resources** → **Storages**, click **Add storage** and edit the settings. In the **ClickHouse cluster nodes** section, specify roles for each server that you are adding: keeper, shard, replica. For more details, see Creating a resource set for a storage.

         The created resource set for the storage is displayed in the **Resources** → **Storages** section. Now you can create storage services for each ClickHouse cluster node.

      2. To create a storage service, in the KUMA web interface, in the **Resources** → **Active services** section, click **Add service**.

         This opens the **Choose a service** window; in that window, select the resource set that you created for the storage at the previous step and click **Create service**. Do the same for each ClickHouse storage.

         The number of created services must be the same as the number of nodes in the ClickHouse cluster, for example, 50 services for 50 nodes. The created storage services are displayed in the KUMA web interface in the **Resources** → **Active services** section. Now you need to install storage services on each node of the ClickHouse cluster by using the service ID.

   b. On the target computer, create the server part of the storage:

      1. In the KUMA web interface, in the **Resources** → **Active services** section, select the storage service that you need and click **Copy ID**.

         The service ID is copied to the clipboard; you need it for running the service installation command.

2. Compose and run the following command on the target computer:

```
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 web interface> --install
```

The storage service is installed on the target computer. You can check the status of the service in the web interface under **Resources** → **Active services**.

3. Run the storage service installation command on each target computer listed in the `storage` section of the `expand.inventory.yml` inventory file, one computer at a time. On each computer, the unique service ID within the cluster must be specified in the installation command.

Dedicated keepers are automatically started immediately after installation and are displayed in the **Resources** → **Active services** section with the green status. Services on other storage nodes may not start until services are installed for all nodes in that cluster. Up to that point, services can be displayed with the red status. This is normal behavior when creating a new storage cluster or adding nodes to an existing storage cluster. As soon as the service installation command is run on all nodes of the cluster, all services get the green status.

7. Specify the added servers in the `distributed.inventory.yml` inventory file so that it has up-to-date information in case of a KUMA update.

The extra storage cluster is successfully added.

- **Removing servers from a distributed installation**.

If necessary, you can remove the server from the distributed installation.

To remove a server from a distributed installation:

1. Remove all services from the server that you want to remove from the distributed installation.
2. Remove the servers from the `distributed.inventory.yml` inventory file to make sure the inventory file has up-to-date information in case you need to upgrade KUMA.

The server is removed from the distributed installation.

- **Removing a storage cluster from a distributed installation**.

You can remove the storage cluster if necessary.

To remove one or more storage clusters from a distributed installation:

1. Get the ID of the storage service. To do this, in the **Resources** → **Active services** section, open the context menu of the service that you are deleting and click **Copy ID**.
2. Remove the storage service on each cluster server that you want to remove from the distributed installation:
   a. Uninstall the server part of the storage service by running the following command on the target machine:

```
sudo /opt/kaspersky/kuma/kuma <storage> --id <service ID> --uninstall
```

Repeat for each server.

b. Remove the client part of the service in the KUMA web interface in the **Resources** → **Active services** → **Delete** section.

The service is removed.

3. Remove servers from the `storage` section of the `distributed.inventory.yml` inventory file to make sure the inventory file has up-to-date information in case you need to upgrade KUMA or modify its configuration.

The cluster is removed from the distributed installation.

- **Migrating the KUMA Core to a new Kubernetes cluster**.

To migrate the KUMA Core to a new Kubernetes cluster:

1. Prepare the k0s.inventory.yml inventory file.

The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the KUMA Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed KUMA Core on all hosts where you plan to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

If the component is not detected on the worker nodes, a clean installation of the KUMA Core is performed in the cluster without migrating resources to it. Existing components must be manually rebuilt with the new Core in the KUMA web interface.

Certificates for collectors, correlators and storages will be re-issued from the inventory file for communication with the Core within the cluster. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes systemd services from the host: kuma-core, kuma-vmalert.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.
- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log

- /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
- /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin

- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed is entered into the ConfigMap.

Updating previous versions of KUMA

The upgrade procedure is the same for all hosts and involves using the installer and inventory file.

Version upgrade scheme:

2.0.x → 2.1.3 → 3.0.3 → 3.2.x → 3.4.x → 4.0

2.1.x → 2.1.3 → 3.0.3 → 3.2.x → 3.4.x → 4.0

2.1.3 → 3.0.3 → 3.2.x → 3.4.x → 4.0

3.0.x → 3.0.3 → 3.2.x → 3.4.x → 4.0

If you made manual changes to the unit files of KUMA services in the /usr/lib/systemd/system directory, these changes are lost when upgrading from a previous version of KUMA. If necessary, make these changes again once the solution is upgraded.

Upgrading from version 2.0.x to 2.1.3

To install KUMA version 2.1.3 over version 2.0.x, complete the preliminary steps and then perform the upgrade.

Preliminary steps

1. Create a backup copy of the KUMA Core. If necessary, you will be able to recover from a backup copy for version 2.0.

KUMA backups created in versions 2.0 and earlier cannot be restored in version 2.1.3. This means that you cannot install KUMA 2.1.3 and restore a KUMA 2.0 backup in it.

Create a backup copy immediately after upgrading KUMA to version 2.1.3.

2. Make sure that all solution installation requirements are met.

3. Make sure that MongoDB versions are compatible by running the following commands on the KUMA Core device:

   ```
   cd /opt/kaspersky/kuma/mongodb/bin/
   ./mongo
   use kuma
   db.adminCommand({getParameter: 1, featureCompatibilityVersion: 1})
   ```

   If the component version is different from 4.4, set the value to 4.4 using the following command:

   ```
   db.adminCommand({ setFeatureCompatibilityVersion: "4.4" })
   ```

4. During installation or upgrade, make sure that TCP port 7220 on the KUMA Core is accessible from the KUMA storage hosts.

5. If you have a keeper deployed on a separate device in the ClickHouse cluster, install the storage service on the same device before you start the upgrade:

   - Use the existing storage of the cluster to create a storage service for the keeper in the web interface.
   - Install the service on the device with the dedicated ClickHouse keeper.

6. In the inventory file, specify the same hosts that were used when installing KUMA version 2.0.X. Set the following settings to `false`:

   ```
   deploy_to_k8s false
   need_transfer false
   deploy_example_services false
   ```

   When the installer uses this inventory file, all KUMA components are upgraded to version 2.1.3. The available services and storage resources are also reconfigured on hosts from the kuma_storage group:

   - ClickHouse systemd services are deleted.
   - Certificates are deleted from the /opt/kaspersky/kuma/clickhouse/certificates directory.
   - The 'Shard ID', 'Replica ID', 'Keeper ID', and 'ClickHouse configuration override' fields are filled in for each node in the storage resource based on values from the inventory file and service configuration files on the host. Subsequently, you will manage the roles of each node in the KUMA web interface.
   - All existing configuration files from the /opt/kaspersky/kuma/clickhouse/cfg directory are deleted (subsequently, they will be generated by the storage service).
   - The value of the LimitNOFILE parameter ('Service' section) is changed from 64,000 to 500,000 in the kuma-storage systemd services.

7. If you use alert segmentation rules, prepare the data for migrating the existing rules and save. In the next step, you can use this data to re-create the rules. During the upgrade, alert segmentation rules are not migrated automatically.

8. To perform an upgrade, you need a valid password from the admin user. If you forgot the password of the admin user, contact Technical Support to reset the current password, then use the new password to perform the upgrade at the next step.

Upgrading KUMA

1. Depending on the KUMA deployment scheme that you are using, do one the following:

- Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the solution.

- Use the prepared k0s.inventory.yml inventory file and follow the instructions for distributed installation in a high availability configuration.

  If an inventory file is not available for the current version, use the provided inventory file template and edit it as necessary. To view a list of hosts and host roles in your current KUMA system, in the web interface, go to **Resources** → **Active services** section.

  The upgrade process mirrors the installation process.

  If you want to upgrade from a distributed installation to a distributed installation in a high availability configuration, you must first upgrade the distributed installation and then **migrate the Core to a Kubernetes cluster**.

  To migrate KUMA Core to a new Kubernetes cluster:

  1. Prepare the k0s.inventory.yml inventory file.

     The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

  2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the KUMA Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed KUMA Core on all hosts where you plan to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

If the component is not detected on the worker nodes, a clean installation of the KUMA Core is performed in the cluster without migrating resources to it. Existing components must be manually rebuilt with the new Core in the KUMA web interface.

Certificates for collectors, correlators and storages will be re-issued from the inventory file for communication with the Core within the cluster. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes systemd services from the host: kuma-core, kuma-vmalert.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.
- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed is entered into the ConfigMap.

1. When upgrading on systems that contain large amounts of data and are operating with limited resources, the system may return the 'Wrong admin password' error message after you enter the administrator password. If you specify the correct password, KUMA may still return this error because of KUMA being unable to start the Core service due to a timeout error and resource limitations. If you enter the administrator password three times without waiting for the installation to complete, the update may end with a fatal error. Resolve the timeout error to proceed with the update.

The final stage of preparing KUMA for work

1. After updating KUMA, you must clear your browser cache.
2. Re-create the alert segmentation rules.
3. Manually upgrade the KUMA agents.

KUMA is successfully upgraded.

Upgrading from version 2.1.x to 2.1.3

To install KUMA version 2.1.3 over version 2.1.x, complete the preliminary steps and then perform the upgrade.

Preliminary steps

1. Creating a backup copy of the KUMA Core. If necessary, you will be able to recover from a backup copy for version 2.1.x.

   KUMA backups created in versions earlier than 2.1.3 cannot be restored in version 2.1.3. This means that you cannot install KUMA 2.1.3 and restore a KUMA 2.1.x backup in it.

   Create a backup copy immediately after upgrading KUMA to version 2.1.3.
2. Make sure that all solution installation requirements are met.
3. During installation or update, ensure network accessibility of TCP port 7220 on the KUMA Core for the KUMA storage hosts.
4. To perform an update, you need a valid password from the admin user. If you forgot the password of the admin user, contact Technical Support to reset the current password, then use the new password to perform the upgrade at the next step.

Upgrading KUMA

1. Depending on the KUMA deployment scheme that you are using, do one the following:

   - Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the solution.
   - Use the prepared k0s.inventory.yml inventory file and follow the instructions for distributed installation in a high availability configuration.

   If an inventory file is not available for the current version, use the provided inventory file template and edit it as necessary. To view a list of hosts and host roles in your current KUMA system, in the web interface, go to **Resources** → **Active services** section.

   The upgrade process mirrors the installation process.

If you want to upgrade from a distributed installation to a distributed installation in a high availability configuration, you must first upgrade the distributed installation and then **migrate the Core to a Kubernetes cluster**.

To migrate KUMA Core to a new Kubernetes cluster:

1. Prepare the k0s.inventory.yml inventory file.

   The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the KUMA Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed KUMA Core on all hosts where you plan to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

If the component is not detected on the worker nodes, a clean installation of the KUMA Core is performed in the cluster without migrating resources to it. Existing components must be manually rebuilt with the new Core in the KUMA web interface.

Certificates for collectors, correlators and storages will be re-issued from the inventory file for communication with the Core within the cluster. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes systemd services from the host: kuma-core, kuma-vmalert.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.
- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.

- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

  After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

  If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

  If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

  If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed is entered into the ConfigMap.

2. When upgrading on systems that contain large amounts of data and are operating with limited resources, the system may return the 'Wrong admin password' error message after you enter the administrator password. If you specify the correct password, KUMA may still return this error because of KUMA being unable to start the Core service due to a timeout error and resource limitations. If you enter the administrator password three times without waiting for the installation to complete, the update may end with a fatal error. Resolve the timeout error to proceed with the update.

The final stage of preparing KUMA for work

1. After updating KUMA, you must clear your browser cache.
2. Manually update the KUMA agents.

KUMA update completed successfully.

Upgrading from version 2.1.3 to 3.0.3

To install KUMA version 3.0.3 over version 2.1.3, complete the preliminary steps and then perform the upgrade.

Preliminary steps

1. Creating a backup copy of the KUMA Core. If necessary, you will be able to restore data from backup for version 3.0.3.

   KUMA backups created in versions 2.1.3 and earlier cannot be restored in version 3.0.3. This means that you cannot install KUMA 3.0.3 and restore a KUMA 2.1.3 backup in it.

   Create a backup copy immediately after upgrading KUMA to version 3.0.3.
2. Make sure that all solution installation requirements are met.
3. During installation or update, ensure network accessibility of TCP port 7220 on the KUMA Core for the KUMA storage hosts.

Upgrading KUMA

Depending on the KUMA deployment scheme that you are using, do one the following:

- Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the solution.
- Use the prepared k0s.inventory.yml inventory file and follow the instructions for distributed installation in a high availability configuration.

  If an inventory file is not available for the current version, use the provided inventory file template and edit it as necessary. To view a list of hosts and host roles in your current KUMA system, in the web interface, go to **Resources** → **Active services** section.

The upgrade process mirrors the installation process.

If you want to upgrade from a distributed installation to a distributed installation in a high availability configuration, you must first upgrade the distributed installation and then **migrate the Core to a Kubernetes cluster**.

To migrate KUMA Core to a new Kubernetes cluster:

1. Prepare the k0s.inventory.yml inventory file.

   The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the KUMA Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed KUMA Core on all hosts where you plan to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

If the component is not detected on the worker nodes, a clean installation of the KUMA Core is performed in the cluster without migrating resources to it. Existing components must be manually rebuilt with the new Core in the KUMA web interface.

Certificates for collectors, correlators and storages will be re-issued from the inventory file for communication with the Core within the cluster. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes systemd services from the host: kuma-core, kuma-vmalert.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.

- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed are entered into the ConfigMap.

The final stage of preparing KUMA for work

1. After updating KUMA, you must clear your browser cache.
2. Manually update the KUMA agents.

KUMA update completed successfully.

Known limitations

1. Hierarchical structure is not supported in 3.0.2, therefore all KUMA hosts become standalone hosts when upgrading from version 2.1.3 to 3.0.2.
2. For existing users, after upgrading from 2.1.3 to 3.0.2, the universal dashboard layout is not refreshed.

   Possible solution: restart the Core service (kuma-core.service), and the data will be refreshed with the interval configured for the layout.

Upgrading from version 3.0.x to 3.0.3

To install KUMA version 3.0.3 over version 3.0.x, complete the preliminary steps and then perform the upgrade.

Preliminary steps

1. Creating a backup copy of the KUMA Core. If necessary, you will be able to restore data from backup for version 3.0.x.

   KUMA backups created in versions earlier than 3.0.3 cannot be restored in version 3.0.3. This means that you cannot install KUMA 3.0.3 and restore a KUMA 3.0.x backup in it.

   Create a backup copy immediately after upgrading KUMA to version 3.0.3.

2. Make sure that all solution installation requirements are met.

3. During installation or update, ensure network accessibility of TCP port 7220 on the KUMA Core for the KUMA storage hosts.

Updating KUMA

Depending on the KUMA deployment scheme that you are using, do one the following:

- Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the solution.

- Use the prepared k0s.inventory.yml inventory file and follow the instructions for distributed installation in a high availability configuration.

  If an inventory file is not available for the current version, use the provided inventory file template and edit it as necessary. To view a list of hosts and host roles in your current KUMA system, in the web interface, go to **Resources** → **Active services** section.

The upgrade process mirrors the installation process.

If you want to upgrade from a distributed installation to a distributed installation in a high availability configuration, you must first upgrade the distributed installation and then **migrate the Core to a Kubernetes cluster**.

To migrate KUMA Core to a new Kubernetes cluster:

1. Prepare the k0s.inventory.yml inventory file.

   The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the KUMA Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed KUMA Core on all hosts where you plan to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

If the component is not detected on the worker nodes, a clean installation of the KUMA Core is performed in the cluster without migrating resources to it. Existing components must be manually rebuilt with the new Core in the KUMA web interface.

Certificates for collectors, correlators and storages will be re-issued from the inventory file for communication with the Core within the cluster. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes systemd services from the host: kuma-core, kuma-vmalert.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.
- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed are entered into the ConfigMap.

The final stage of preparing KUMA for work

1. After updating KUMA, you must clear your browser cache.
2. Manually update the KUMA agents.

KUMA update completed successfully.

Known limitations

For existing users, after upgrading from 3.0.x to 3.0.3, the universal dashboard layout is not refreshed.

Possible solution: restart the Core service (kuma-core.service), and the data refresh interval configured for the layout will be used.

Upgrading from version 3.0.3 to 3.2.x

To install KUMA version 3.2.x over version 3.0.3, complete the preliminary steps and then perform the upgrade.

Preliminary steps

1. Creating a backup copy of the KUMA Core. If necessary, you can restore data from backup for version 3.0.3.

   KUMA backups created in versions 3.0.3 and earlier cannot be restored in version 3.2.x. This means that you cannot install KUMA 3.2.x and restore a KUMA 3.0.3 backup in it.

   Create a backup copy immediately after upgrading KUMA to version 3.2.x.
2. Make sure that all solution installation requirements are met.
3. Make sure that the host name of the KUMA Core does not start with a numeral. The upgrade to version 3.2.x cannot be completed successfully if the host name of the KUMA Core starts with a numeral. In such a case, you will need to take certain measures to successfully complete the upgrade. Contact Technical Support for additional instructions.
4. During installation or update, ensure network accessibility of TCP port 7220 on the KUMA Core for the KUMA storage hosts.

Updating KUMA

Depending on the KUMA deployment scheme that you are using, do one the following:

- Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the solution.
- Use the prepared k0s.inventory.yml inventory file and follow the instructions for distributed installation in a high availability configuration.

  If an inventory file is not available for the current version, use the provided inventory file template and edit it as necessary. To view a list of hosts and host roles in your current KUMA system, in the web interface, go to **Resources** → **Active services** section.

The upgrade process mirrors the installation process.

If you want to upgrade from a distributed installation to a distributed installation in a high availability configuration, first upgrade the distributed installation and then **migrate the Core to a Kubernetes cluster**. For subsequent upgrades, use the k0s.inventory.yml inventory file with the need_transfer parameter set to false, because the KUMA Core already has been migrated to the Kubernetes cluster and you do not need to repeat this.

To migrate KUMA Core to a new Kubernetes cluster:

1. Prepare the k0s.inventory.yml inventory file.

   The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the KUMA Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed KUMA Core on all hosts where you plan to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

If the component is not detected on the worker nodes, a clean installation of the KUMA Core is performed in the cluster without migrating resources to it. Existing components must be manually rebuilt with the new Core in the KUMA web interface.

Certificates for collectors, correlators and storages will be re-issued from the inventory file for communication with the Core within the cluster. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes systemd services from the host: kuma-core, kuma-vmalert.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.

- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed are entered into the ConfigMap.

The final stage of preparing KUMA for work

1. After updating KUMA, you must clear your browser cache.
2. If you are using agents, manually update the KUMA agents.

KUMA update completed successfully.

Known limitations

1. For existing users, after upgrading from 3.0.3 to 3.2.x, the universal dashboard layout is not refreshed.

   Possible solution: restart the Core service (kuma-core.service), and the data refresh interval configured for the layout will be used.

2. If the old Core service, "kuma-core.service" is still displayed after the upgrade, run the following command after installation is complete:

sudo systemctl reset-failed

After running the command, the old service is no longer displayed, and the new service starts successfully.

Upgrading from version 3.2.0 or 3.2.1 to 3.4

To install KUMA version 3.4 over version 3.2.0 or 3.2.1, complete the preliminary steps and then perform the upgrade.

Preliminary steps

1. Creating a backup copy of the KUMA Core. If necessary, you will be able to restore data from backup for version 3.2.

   KUMA backups created in versions 3.2 and earlier cannot be restored in version 3.4. This means that you cannot install KUMA 3.4 and restore a KUMA 3.2.x backup in it.

   Create a backup copy immediately after upgrading KUMA to version 3.4.

2. Make sure that all solution installation requirements are met.

3. Make sure that the host name of the KUMA Core does not start with a numeral. The upgrade to version 3.4 cannot be completed successfully if the host name of the KUMA Core begins with a numeral. If the hostname of the KUMA Core starts with a numeral, a number of steps are necessary to successfully complete the upgrade. Contact Technical Support for additional instructions.

4. During installation or update, ensure network accessibility of TCP port 7220 on the KUMA Core for the KUMA storage hosts.

5. For event sources that have a red status in the **Source status** → **List of event sources** section, check out how many there are and if they are up to date; for outdated event sources, do one of the following:

   - Remove the event sources that you no longer need.
   - Edit the settings of monitoring policies that are applied to the outdated event sources to bring them up to date with the current stream of events.
   - For monitoring policies applied to outdated event sources, disable notifications.

   After KUMA is upgraded, notifications are sent to the configured email addresses for all red-status event sources that have notifications about triggered policies configured.

Updating KUMA

Depending on the KUMA deployment scheme that you are using, do one the following:

- Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the solution.
- Use the prepared k0s.inventory.yml inventory file and follow the instructions for distributed installation in a high availability configuration.

  If an inventory file is not available for the current version, use the provided inventory file template and edit it as necessary. To view a list of hosts and host roles in your current KUMA system, in the web interface, go to **Resources** → **Active services** section.

The upgrade process mirrors the installation process.

If you want to upgrade from a distributed installation to a distributed installation in a high availability configuration, first upgrade the distributed installation and then **migrate the Core to a Kubernetes cluster**. For subsequent upgrades, use the k0s.inventory.yml inventory file with the need_transfer parameter set to false, because the KUMA Core already has been migrated to the Kubernetes cluster and you do not need to repeat this.

To migrate KUMA Core to a new Kubernetes cluster:

1. Prepare the k0s.inventory.yml inventory file.

   The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the KUMA Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed KUMA Core on all hosts where you plan to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

If the component is not detected on the worker nodes, a clean installation of the KUMA Core is performed in the cluster without migrating resources to it. Existing components must be manually rebuilt with the new Core in the KUMA web interface.

Certificates for collectors, correlators and storages will be re-issued from the inventory file for communication with the Core within the cluster. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes systemd services from the host: kuma-core, kuma-vmalert.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.

- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed are entered into the ConfigMap.

The final stage of preparing KUMA for work

1. After updating KUMA, you must clear your browser cache.
2. If you are using agents, manually update the KUMA agents.

KUMA update completed successfully.

Known limitations

1. For existing users, after upgrading from 3.2.0 or 3.2.1 to 3.4, the universal dashboard layout is not refreshed.

   Possible solution: restart the Core service (kuma-core.service), and the data refresh interval configured for the layout will be used.

2. If the old Core service, "kuma-core.service" is still displayed after the upgrade, run the following command after installation is complete:

sudo systemctl reset-failed

After running the command, the old service is no longer displayed, and the new service starts successfully.

Upgrading from version 3.4.x to 4.0

To install KUMA version 4.0 over version 3.4.x, complete the preliminary steps and then perform the upgrade.

Preliminary steps

1. Creating a backup copy of the KUMA Core. If necessary, you will be able to restore data from backup for version 3.4.x.

   KUMA backups created in versions 3.4.x and earlier cannot be restored in version 4.0. This means that you cannot install KUMA 4.0 and restore a KUMA 3.4.x backup in it.

   Create a backup copy immediately after upgrading KUMA to version 4.0.

2. Make sure that all solution installation requirements are met.

3. Make sure that the host name of the KUMA Core does not start with a numeral. The upgrade to version 4.0 cannot be completed successfully if the host name of the KUMA Core begins with a numeral. If the hostname of the KUMA Core starts with a numeral, a number of steps are necessary to successfully complete the upgrade. Contact Technical Support for additional instructions.

4. During installation or update, ensure network accessibility of TCP port 7220 on the KUMA Core for the KUMA storage hosts.

5. For event sources that have a red status in the **Source status** → **List of event sources** section, check out how many there are and if they are up to date; for outdated event sources, do one of the following:

   - Remove the event sources that you no longer need.
   - Edit the settings of monitoring policies that are applied to the outdated event sources to bring them up to date with the current stream of events.
   - For monitoring policies applied to outdated event sources, disable notifications.

   After KUMA is upgraded, notifications are sent to the configured email addresses for all red-status event sources that have notifications about triggered policies configured.

6. For an installation with the KUMA Core in a Kubernetes cluster, run the script to determine the size of metrics data on the KUMA Core 3.4 volume; if necessary, set or modify the values of `metrics_transfer_job_timeout`, `metrics_disk_request`, `core_disk_request` parameters in the k0s.inventory.yml inventory file; and then proceed with the update procedure.

Upgrading KUMA

Depending on the KUMA deployment scheme that you are using, do one the following:

- Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the solution.
- Use the prepared k0s.inventory.yml inventory file and follow the instructions for distributed installation in a high availability configuration with the KUMA Core in a Kubernetes cluster.

  If an inventory file is not available for the current version, use the provided inventory file template and fill in the corresponding settings. To view a list of hosts and host roles in your current KUMA system, in the web interface, go to **Resources** → **Active services** section.

The upgrade process mirrors the installation process.

You can change the installation type as follows:

- If you want to upgrade from a distributed installation to an installation with the KUMA Core in a Kubernetes cluster, you must first upgrade the distributed installation to the latest version and then **migrate the Core to a Kubernetes cluster**. For subsequent upgrades, use the k0s.inventory.yml inventory file with `need_transfer: false`, because the KUMA Core already has been migrated to the Kubernetes cluster and you do not need to repeat this.

  To migrate KUMA Core to a new Kubernetes cluster:

  1. Prepare the k0s.inventory.yml inventory file.

     The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

  2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the KUMA Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed KUMA Core on all hosts where you plan to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

If the component is not detected on the worker nodes, a clean installation of the KUMA Core is performed in the cluster without migrating resources to it. Existing components must be manually rebuilt with the new Core in the KUMA web interface.

Certificates for collectors, correlators and storages will be re-issued from the inventory file for communication with the Core within the cluster. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes systemd services from the host: kuma-core, kuma-vmalert.
- Deletes the internal certificate of the Core.

- Deletes the certificate files of all other components and deletes their records from MongoDB.
- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

  After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

  If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

  If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

  If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed are entered into the ConfigMap.

- If you want to upgrade KUMA in a distributed installation to the latest version of KUMA with the Core in a Raft cluster, you must manually add a `kuma_core_peers` group to the distributed.inventory.yml inventory file and in this group, specify an even number of servers on which you plan to install additional KUMA Core services. Also, in the `kuma_core` and `kuma_core_peers` groups, you can specify the `raft_node_addr` setting in the `<FQDN of the server from the kuma_core group on which the KUMA Core is installed>:<port>` format, if the server has multiple FQDNs. The FQDN specified in the `raft_node_addr` setting will be used to listen for signals from other nodes.

  After starting the installer with the prepared inventory file, none of the additional KUMA Core services are automatically installed on the servers. The installer only prepares the servers specified in the inventory file for the installation of services. After the installer finishes, only one KUMA Core service is automatically installed on the server specified in the `kuma_core` group. After the installer finishes, you need to manually create the KUMA Core services in the web interface and install the KUMA Core services with unique IDs one by one on each server specified in the `kuma_core_peers` group.

  After the upgrade, all existing KUMA services are connected to the KUMA Core on the server from the `kuma_core` group. Subsequently, when installing new services, you can specify multiple comma-separated URLs in the installation command to connect services to KUMA Cores in the Raft cluster.

The final stage of preparing KUMA for work

1. After updating KUMA, you must clear your browser cache.
2. If you are using agents, manually update the KUMA agents.
3. Migrate alert events from MongoDB to ClickHouse.
4. Specify the storage for new alert events in alert filling settings.

KUMA update completed successfully.

Known limitations

1. For existing users, upgrading from 3.4.x to 4.0 does not update the universal dashboard.

   Possible solution: restart the Core service (kuma-core.service) to apply the data refresh interval configured for the dashboard.

2. If the old Core service, "kuma-core.service" is still displayed after the upgrade, run the following command after installation is complete:

   ```
   sudo systemctl reset-failed
   ```

   After running the command, the old service is no longer displayed, and the new service starts successfully.

Upgrading from version 4.0.x to 4.2

To install KUMA version 4.2 over version 4.0.x, complete the preliminary steps and then perform the upgrade.

Preliminary steps

1. Creating a backup copy of the KUMA Core. If necessary, you will be able to restore data from backup for version 4.0.x.

   KUMA backups created in versions 4.0.x and earlier cannot be restored in version 4.2 This means that you cannot install KUMA 4.2 and restore a KUMA 4.0.x backup in it.

   Create a backup copy immediately after upgrading KUMA to version 4.2.

2. Make sure that all solution installation requirements are met.
3. Make sure that the host name of the KUMA Core does not start with a numeral. The upgrade to version 4.2 cannot be completed successfully if the host name of the KUMA Core begins with a numeral. If the hostname of the KUMA Core starts with a numeral, a number of steps are necessary to successfully complete the upgrade. Contact Technical Support for additional instructions.
4. During installation or update, ensure network accessibility of TCP port 7220 on the KUMA Core for the KUMA storage hosts.

5. For event sources that have a red status in the **Source status** → **List of event sources** section, check out how many there are and if they are up to date; for outdated event sources, do one of the following:

   - Remove the event sources that you no longer need.
   - Edit the settings of monitoring policies that are applied to the outdated event sources to bring them up to date with the current stream of events.
   - For monitoring policies applied to outdated event sources, disable notifications.

   After KUMA is upgraded, notifications are sent to the configured email addresses for all red-status event sources that have notifications about triggered policies configured.

Updating KUMA

Depending on the KUMA deployment scheme that you are using, do one the following:

- Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the solution. In the inventory file, in the `kuma_metrics` section, you need to specify the host on which the active Metrics service is running. When the update starts, a check is performed, and if the Metrics service is not found on the specified host, an error is returned and the update is stopped.
- Use the prepared k0s.inventory.yml inventory file and follow the instructions for distributed installation in a high availability configuration with the KUMA Core in a Kubernetes cluster. In the inventory file, in the `kuma_metrics` section, you need to specify the host on which the active Metrics service is running. When the update starts, a check is performed, and if the Metrics service is not found on the specified host, an error is returned and the update is stopped.

  If an inventory file is not available for the current version, use the provided inventory file template and fill in the corresponding settings. To view a list of hosts and host roles in your current KUMA system, in the web interface, go to **Resources** → **Active services** section.

The upgrade process mirrors the installation process.

You can change the installation type as follows:

- If you want to upgrade from a distributed installation to an installation with the KUMA Core in a Kubernetes cluster, you must first upgrade the distributed installation to the latest version and then **migrate the Core to a Kubernetes cluster**. For subsequent upgrades, use the k0s.inventory.yml inventory file with `need_transfer: false`, because the KUMA Core already has been migrated to the Kubernetes cluster and you do not need to repeat this.

  To migrate KUMA Core to a new Kubernetes cluster:

  1. Prepare the k0s.inventory.yml inventory file.

     The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

  2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the KUMA Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed KUMA Core on all hosts where you plan to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

If the component is not detected on the worker nodes, a clean installation of the KUMA Core is performed in the cluster without migrating resources to it. Existing components must be manually rebuilt with the new Core in the KUMA web interface.

Certificates for collectors, correlators and storages will be re-issued from the inventory file for communication with the Core within the cluster. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes systemd services from the host: kuma-core, kuma-vmalert.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.
- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed are entered into the ConfigMap.

- If you want to upgrade KUMA in a distributed installation to the latest version of KUMA with the Core in a Raft cluster, you must manually add a `kuma_core_peers` group to the distributed.inventory.yml inventory file and in this group, specify an even number of servers on which you plan to install additional KUMA Core services. Also, in the `kuma_core` and `kuma_core_peers` groups, you can specify the `raft_node_addr` setting in the `<FQDN of the server from the kuma_core group on which the KUMA Core is installed>:<port>` format, if the server has multiple FQDNs. The FQDN specified in the `raft_node_addr` setting will be used to listen for signals from other nodes.

  After starting the installer with the prepared inventory file, none of the additional KUMA Core services are automatically installed on the servers. The installer only prepares the servers specified in the inventory file for the installation of services. After the installer finishes, only one KUMA Core service is automatically installed on the server specified in the `kuma_core` group. After the installer finishes, you need to manually create the KUMA Core services in the web interface and install the KUMA Core services with unique IDs one by one on each server specified in the `kuma_core_peers` group.

  After the upgrade, all existing KUMA services are connected to the KUMA Core on the server from the `kuma_core` group. Subsequently, when installing new services, you can specify multiple comma-separated URLs in the installation command to connect services to KUMA Cores in the Raft cluster.

The final stage of preparing KUMA for work

1. After updating KUMA, you must clear your browser cache.
2. If you are using agents, manually update the KUMA agents.
3. For a distributed installation in a high-availability configuration with KUMA Core in a Kubernetes cluster, upgrade the cluster components.

KUMA update completed successfully.

Known limitations

1. For existing users, upgrading from 4.0.x to 4.2 does not update the universal dashboard.

   Possible solution: restart the Core service (kuma-core-`<KUMA Core service ID>`.service) to apply the data refresh interval configured for the dashboard.

Upgrading components of the Kubernetes cluster

Kubernetes cluster components need to be upgraded only if KUMA is being upgraded to version 4.2.x. When installing KUMA 4.2.x from scratch, a Kubernetes cluster component upgrade is not necessary.

Before upgrading your Kubernetes cluster components, you need to manually back up the KUMA Core 4.2 in case of unforeseen issues with the upgrade. If you do not have a manually created backup copy of the KUMA Core 4.2, all data may be lost in the event of a failure.

Kubernetes cluster components must be upgraded from a dedicated ansible-controller machine that is not a member of the Kubernetes cluster because the upgrade script restarts all cluster hosts. If a machine inside the cluster is used as the ansible-controller, the restart of the cluster hosts stops the execution of the script and the cluster becomes inoperable.

Kubernetes cluster components are upgraded using the cluster_upgrade.sh script and the k0s.inventory.yaml inventory file, only on hosts from the `k0s` section of the k0s.inventory.yaml inventory file. External KUMA services are not affected.

Logic of the cluster_upgrade.sh script

By default, the upgrade script automatically creates a backup copy.

If creating a manual backup of the KUMA Core, as recommended for unforeseen contingencies, took a long time and you do not want to wait a similar amount of time for the automatic backup to be created, you can place your manual backup on the dedicated ansible-controller machine in a file named autobackup.tar.gz file and run the script with the `-e 'manual_restore_only=true'` option. The `-e 'manual_restore_only=true'` option skips the automatic backup and restore, speeding up the upgrade. In this case, you will need to restore the KUMA Core from a backup copy manually.

The following methods are available for upgrading Kubernetes cluster components:

- Upgrading Kubernetes cluster components without the -e 'manual_restore_only=true' option
- Upgrading Kubernetes cluster components with the -e 'manual_restore_only=true' option

Upgrading Kubernetes cluster components without the -e 'manual_restore_only=true' option

To upgrade Kubernetes cluster components without the -e 'manual_restore_only=true' option

1. Prepare for the upgrade:

   a. Make sure that KUMA has been upgraded from version 4.0.x to 4.2.x using the install.sh installer and the k0s.inventory.yaml inventory file.

   b. Prepare the dedicated ansible-controller machine from which you are going to run the update script, and make sure that this machine does not belong to the cluster. The update script restarts all hosts in the cluster. If the ansible-controller machine belongs to the cluster, the restart of the cluster hosts stops the script, and the installation of new versions of Kubernetes components fails, leaving the cluster inoperable.

   c. Make sure there is enough free disk space on the dedicated ansible-controller machine and on the master-controller to accommodate the backup data.

   d. Manually create a backup copy of the KUMA Core to prepare for unforeseen issues.

      The metrics data is not included in the backup copy and is not migrated to the Kubernetes cluster. When you upgrade the components of the Kubernetes cluster, the metrics data will be lost.

      If the metrics data is critically important to you, contact Technical Support before upgrading the Kubernetes cluster components.

   e. Specify the `backup_timeout` setting in the k0s.inventory.yaml inventory file in the `vars` section.

2. Perform the upgrade:

   a. On the dedicated ansible-controller machine, go to the directory of the 4.2 version installer and run the cluster_upgrade.sh script, passing the name of the current inventory file k0s.inventory.yaml as an argument:

   ```
   ./cluster_upgrade.sh k0s.inventory.yaml
   ```

   b. On command line, confirm that you have a backup copy of the KUMA Core and agree to the upgrade by entering 'yes'.

   During the upgrade process, the KUMA Core is automatically backed up in the autobackup.tar.gz archive. The script creates a backup copy of the KUMA Core on the master-controller machine in the /root/k0s directory, and then transfers it to the directory with the script on the dedicated ansible-controller machine from which you are upgrading the Kubernetes components. The automatic recovery of KUMA Core data in the new Kubernetes cluster also involves the installer directory and the /root/k0s directory on the master-controller machine.

   Later, the KUMA Core will be automatically restored from the backup copy in the new Kubernetes cluster. If the upgrade fails, you can place your manually created backup copy into the autobackup.tar.gz file in the script directory on the ansible-controller machine and run the cluster_upgrade.sh script again.

   The created backup copy is not automatically overwritten.

Components of the Kubernetes cluster are successfully upgraded.

Failure recovery

The following scenarios are available for failure recovery:

- Failure to upgrade Kubernetes cluster components. You can use any of the following recovery options:
  - Place your manually created backup copy into the autobackup.tar.gz file in the script directory on the master-controller machine and run the cluster_upgrade.sh script again. The script processes the manually created backup copy and continues the upgrade procedure.
  - Run the cluster_upgrade.sh script with the `-e 'manual_restore_only=true'` option and perform the recovery manually.
- Failure to start the KUMA Core after a successful upgrade of the Kubernetes cluster.

  If you want the script to automatically repeat the uninstallation and new installation of the cluster, you must stop the k0s service on any of the cluster nodes and delete the k0s executable file on that node. As a result, the script will interpret the state of the cluster as inconsistent and will uninstall and reinstall the cluster using the previously created autobackup.tar.gz file.

  To stop the k0s service on cluster controllers and hosts that combine the roles of a controller and a worker node, run the following commands:

  ```
  sudo systemctl stop k0scontroller.service
  sudo rm /usr/bin/k0s
  ```

  To stop the k0s service and delete the k0s binary file on dedicated worker nodes, run the following commands:

  ```
  sudo systemctl stop k0sworker.service
  sudo rm /usr/bin/k0s
  ```

  The cluster is uninstalled only after obtaining a correct backup file or explicitly refusing to use the automatic backup and restoration by passing the `-e 'manual_restore_only=true'` option and agreeing to proceed with the upgrade without performing a backup. If you do not have a manually created backup copy of the, all data will be lost in the event of a failure.

Upgrading Kubernetes cluster components with the -e 'manual_restore_only=true' option

To upgrade Kubernetes cluster components with the -e 'manual_restore_only=true' option

1. Prepare for the upgrade:

   a. Make sure that KUMA has been upgraded from version 4.0.x to 4.2.x using the install.sh installer and the k0s.inventory.yaml inventory file.

   b. Prepare the dedicated ansible-controller machine from which you are going to run the update script, and make sure that this machine does not belong to the cluster. The update script restarts all hosts in the cluster. If the ansible-controller machine belongs to the cluster, the restart of the cluster hosts stops the script, and the installation of new versions of Kubernetes components fails, leaving the cluster inoperable.

   c. Make sure there is enough free disk space on the dedicated ansible-controller machine and on the master-controller to accommodate the backup data.

   d. Manually create a backup copy of the KUMA Core to prepare for unforeseen issues.

      The metrics data is not included in the backup copy and is not migrated to the Kubernetes cluster. When you upgrade the components of the Kubernetes cluster, the metrics data will be lost.

      If the metrics data is critically important to you, contact Technical Support before upgrading the Kubernetes cluster components.

2. Perform the upgrade:

   a. On the dedicated ansible-controller machine, go to the directory of the KUMA 4.2 installer and run the cluster_upgrade.sh script, passing the name of the current inventory file k0s.inventory.yaml and the `-e 'manual_restore_only=true'` option:

      ```
      ./cluster_upgrade.sh k0s.inventory.yaml [-e 'manual_restore_only=true']
      ```

      Automatic backup and restoration is disabled. The script uninstalls the old cluster and creates a new Kubernetes cluster.

   b. On the command line, confirm that you have a backup copy of the KUMA Core and agree to the upgrade by entering `yes`.

      If you do not have a manually created backup copy, confirming the uninstallation of the cluster causes the data to be lost.

3. Restore the KUMA Core from the backup.

Components of the Kubernetes cluster are successfully upgraded.

Failure recovery

If the upgrade fails, run the cluster_upgrade.sh script again and manually restore the KUMA Core from the backup copy.

Sizing the metrics data in an installation with KUMA Core 3.4.x in a Kubernetes cluster

When upgrading from KUMA 3.4 to 4.0.x, you must determine how much disk space the metrics data occupies and evaluate whether it is necessary to override the `metrics_disk_request` setting in the k0s.inventory.yml inventory file.

To evaluate the amount of disk space that will be taken up by metrics data in KUMA 3.4:

On the target machine, run the get_old_metrics_size.sh script with the k0s.inventory.yml inventory file:

```
sudo ./get_old_metrics_size_sh k0s.inventory.yml
```

This script does not make any changes to the existing installation; it connects to the primary controller of the cluster to determine the name of the KUMA Core volume and to the worker nodes of the cluster to determine the directory size inside the mounted volume.

As a result, a search for KUMA Core volumes is performed on the worker nodes of the cluster, and the size of the victoria-metrics directory in them is determined. The output of the script is displayed and written to the log/get_old_metrics_size-YYYYMMDD-HHMMSS.log file. The following message is logged:

```
TASK [Print metrics folder size] ********************************************
ok: [kuma-inst-06.avp.ru] => {}
MSG:
Found KUMA Core 3.4.1.53
Found old metrics folder size: 1Gi
Minimum metrics_data_size required to proceed: 3Gi (including 2Gi threshold)
```

The message contains the following information:

- `Found KUMA Core` is the found version of the KUMA Core.
- `Found old metrics folder size` is current size of the victoria-metrics directory in GB.
- `Minimum metrics_data_size required to proceed` is the minimum possible size of the metrics volume for migration from version 3.4.x. This size includes a safety margin of 2Gi of free space. If necessary, you can filter the output by the `Minimum metrics_data_size required to proceed` string.

Based on the results, you need to determine the required size of the Metrics service volume in KUMA 4.0.x, depending on your needs. The Metrics service volume size must not be less than the `Minimum metrics_data_size required to proceed` value.

When upgrading from version 3.4.x to version 4.0.x, if you specify a volume size less than the `Minimum metrics_data_size required to proceed` value in the inventory file k0s.inventory.yml, or if you specify the `metrics_disk_request` value that is less than the `Minimum metrics_data_size required to proceed` value, the installer aborts the upgrade at the installation validation stage with the following message:

```
TASK [Check metrics volume size requirement] ********************************
fatal: [kuma-inst-06.avp.ru]: FAILED! => {
"changed": false
}
MSG:
Current metrics folder size: 20Gi
Metrics disk request when low_resources=false: 10Gi
```

Metrics disk request when low_resources=true: 10Gi
Error: Available metric data exceeds specified volume size.
The available metric data (20Gi + 2Gi threshold) is too large.
Solution: Increase the metrics volume size by updating the metrics_disk_request variable in your inventory file.

In the above example, the user tried to set the volume size to 10Gi through the metrics_disk_request inventory variable, while the current size of metrics data on the disk is 20Gi. The intended volume size is less than the current data size + the 2Gi safety margin, therefore the upgrade cannot be performed. The installation is aborted at the validation step, and no changes are made to the current installation. You can safely choose a suitable volume size for metrics, override it in the inventory file, and retry the upgrade.

Growing the KUMA Core volume or the Metrics service volume in an active Kubernetes cluster

You may need to grow a volume for the KUMA Core or Metrics service for a KUMA installation in production.

Before growing volumes, you need to make sure that there is enough free space in the /opt directory on the worker nodes of the cluster. You need at least twice as much free space as the planned volume size.

Starting with KUMA 4.0.1, the installer includes the roles/k0s_prepare/files/k0s-pvc-expand.sh script, which is copied to the /root/k0s/ directory on the primary controller of the cluster. This script lets you grow a volume in a production cluster.

Growing a volume using the k0s-pvc-expand.sh script

You can grow a volume in one of the following ways:

- Go to the /root/k0s/ directory of the master controller and run the k0s-pvc-expand.sh script with the `core` or `metrics` option, depending on which volume you want to apply the action to, and specify the new volume size in gigabytes. The volume size must be specified as an integer number without a suffix.
- Manually copy the k0s-pvc-expand.sh script to any cluster controller to a directory from which you want to run it and run the script.

Example command for growing the Metrics service volume to 100 GB:

```
sudo ./k0s-pvc-expand.sh metrics 100
```

Example command for growing the KUMA Core volume to 200 GB:

```
sudo ./k0s-pvc-expand.sh core 200
```

Script behavior

If the specified volume size is less than or equal to the current size, nothing is done and the script aborts with an error message.

If the new volume size is greater than the current size, the script stops the service that uses the volume (KUMA Core or Metrics). After disconnecting the volume from the pods of the service, the script grows the volume. After resizing the volume, the service is restarted and the expanded volume becomes available to it.

After growing the volume in preparation for an upgrade to later versions of KUMA, you will not need to further specify any volume sizes in the inventory because the KUMA installer will use current volume sizes and ignore values from variables from the inventory file that do not match the current values.

Shrinking the Metrics service volume

Before shrinking the Metrics volume, you need to determine the sufficient volume size based on the metrics in the **Metrics → KUMA Metrics service → Storage → Disk space usage**. Take into account the metrics storage duration and allocate extra disk space if you plan to use additional sources of metrics.

Before shrinking volumes, make sure that no metrics-transfer-<unique ID> task exists.

```
k0s kubectl get jobs -n kuma
```

The command should output `"No resources found in kuma namespace."`. The metrics-transfer task is automatically deleted 1 hour after completion.

If you want to delete it earlier, you can use the following command:

```
k0s kubectl delete job metrics-transfer -n kuma
```

You can shrink a volume in one of the following ways:

- Shrinking a Metrics service volume without data loss
- Shrinking a Metrics service volume with data loss

Shrinking a Metrics service volume without data loss

To shrink a Metrics service volume without data loss:

1. Unmount the Metrics service volume:

   a. Get the name of the volume used by the Metrics service. You will need this volume name to locate volume data and volume replicas. Replica names begin with the volume name.

   ```
   sudo k0s kubectl get pvc metrics -n kuma -o json | jq '.spec.volumeName'
   ```

   b. Stop the Metrics service.

   ```
   sudo k0s kubectl scale deployment/metrics -n kuma --replicas=0
   ```

   c. Make sure that the metrics-<UID> pod of the Metrics service has disappeared (is not present in the command output).

   ```
   sudo k0s kubectl get pods -n kuma
   ```

d. If the pod does not disappear within a few minutes, force its deletion. The pod name must be taken from the output of the previous command.

```
sudo k0s kubectl delete pod metrics-<UID> -n kuma --force
```

e. Make sure that the Metrics service volume is in the detached state.

```
sudo k0s kubectl get volume <volume_name> -n longhorn-system -o json | jq
'.status.state'
```

2. Create a temporary task:

a. Create a temporary task named tmp-metrics-job in the 'kuma' namespace and mount the Metrics service volume.

```
sudo k0s kubectl apply -f - <<EOF
---
apiVersion: batch/v1
kind: Job
metadata:
  name: tmp-metrics-job
  namespace: kuma
spec:
  ttlSecondsAfterFinished: 300
  backoffLimit: 0
  parallelism: 1
  completions: 1
  template:
    spec:
      containers:
        - name: tmp-container
          image: docker.io/library/busybox:1.35.0
          imagePullPolicy: IfNotPresent
          command: ["/bin/sh"]
          args:
            - -c
            - >-
                sleep 36000
          volumeMounts:
            - mountPath: /metrics
              name: metrics
      restartPolicy: Never
      volumes:
        - name: metrics
          persistentVolumeClaim:
            claimName: metrics
EOF
```

b. Determine the name of the worker node on which the temporary task is running.

```
sudo k0s kubectl get pods -n kuma -o wide | grep tmp-metrics
```

c. On this worker node, determine the mount point.

```
sudo lsblk | grep "<volume_name>"
```

d. Change to the found directory, /var/lib/k0s/kubelet/pods/……/mount. The content should look like this:

```
ls -al
total 48
drwxrwsr-x 9 root  65532  4096 Aug 26 10:36 .
drwxr-x--- 3 root  root   4096 Aug 26 11:08 ..
```

```
drwx--S--- 2 65532 65532  4096 Aug 26 10:36 certificates
drwx--S--- 2 65532 65532  4096 Aug 26 10:36 dashboards
drwxrwsr-x 8 root  65532  4096 Aug 26 10:34 data
drwx--S--- 2 65532 65532  4096 Aug 26 10:36 log
drwxrws--- 2 root  65532 16384 Aug 26 10:34 lost+found
drwx--S--- 2 65532 65532  4096 Aug 26 10:36 rules
drwx--S--- 2 65532 65532  4096 Aug 26 10:36 tmp
```

3. Copy the data:

   a. Copy all data from the /var/lib/k0s/kubelet/pods/...../mount directory, except for the 'lost+found' and 'tmp' directories, to a temporary directory, keeping the permissions on directories and files.

   b. After copying, go from the mount point to the / or /home directory to avoid blocking the subsequent unmounting of the volume.

4. Delete the temporary task, pod, PVC, and volume of the Metrics service:

   a. Delete the temporary task with the following command:

   ```
   sudo k0s kubectl delete job tmp-metrics-job -n kuma
   ```

   b. Make sure that the metrics-<UID> pod of the temporary task is not present in the output of the following command:

   ```
   sudo k0s kubectl get pods -n kuma
   ```

   c. If the pod does not disappear within a few minutes, force its deletion. The pod name must be taken from the output of the previous command.

   ```
   sudo k0s kubectl delete pod tmp-metrics-job-<UID> -n kuma --force
   ```

   d. Make sure that the Metrics service volume is in the detached state.

   ```
   sudo k0s kubectl get volume <volume_name> -n longhorn-system -o json | jq
   '.status.state'
   ```

   e. Delete the PVC of the Metrics service.

   ```
   sudo k0s kubectl delete pvc metrics -n kuma
   ```

   f. Make sure the Metrics service volume is not present in the output of the following command:

   ```
   sudo k0s kubectl get volume <volume_name> -n longhorn-system
   ```

   On all worker nodes, directories with names beginning with the name of the deleted volume should disappear from the /opt/longhorn/replicas directory. If this does not happen, you need to delete them manually. Take care not to delete replicas that belong to other volumes by mistake.

5. Create a new volume:

   a. Specify the value in the 'storage' line, then create a PVC for the Metrics service.

   ```
   sudo k0s kubectl apply -f - <<EOF
   ---
   apiVersion: v1
   kind: PersistentVolumeClaim
   metadata:
     name: metrics
     namespace: kuma
   spec:
     accessModes:
   ```

```
    - ReadWriteOnce
  storageClassName: kuma-network-storage
  resources:
    requests:
      storage: 30Gi
EOF
```

b. Get the name of the volume used by the Metrics service again.

```
sudo k0s kubectl get pvc metrics -n kuma -o json | jq '.spec.volumeName'
```

c. Create a temporary task named tmp-metrics-job in the 'kuma' namespace again and mount the Metrics service volume.

```
sudo k0s kubectl apply -f - <<EOF
---
apiVersion: batch/v1
kind: Job
metadata:
  name: tmp-metrics-job
  namespace: kuma
spec:
  ttlSecondsAfterFinished: 300
  backoffLimit: 0
  parallelism: 1
  completions: 1
  template:
    spec:
      containers:
        - name: tmp-container
          image: docker.io/library/busybox:1.35.0
          imagePullPolicy: IfNotPresent
          command: ["/bin/sh"]
          args:
            - -c
            - >-
                sleep 36000
          volumeMounts:
            - mountPath: /metrics
              name: metrics
      restartPolicy: Never
      volumes:
        - name: metrics
          persistentVolumeClaim:
            claimName: metrics
EOF
```

d. Determine the name of the worker node on which the temporary task is running.

```
sudo k0s kubectl get pods -n kuma -o wide | grep tmp-metrics
```

e. On this worker node, determine the mount point.

```
sudo lsblk | grep "<volume_name>"
```

f. Change to the found directory, /var/lib/k0s/kubelet/pods/……/mount.

6. Copy the data:

   a. Copy the Metrics service data previously saved in the temporary directory to the located /var/lib/k0s/kubelet/pods/...../mount directory, keeping the ownership and permissions of the files. As a result, the contents of the directory should look like this.

   ```
   ls -al
   total 48
   drwxrwsr-x 9 root  65532  4096 Aug 26 10:36 .
   drwxr-x--- 3 root  root   4096 Aug 26 11:08 ..
   drwx--S--- 2 65532 65532  4096 Aug 26 10:36 certificates
   drwx--S--- 2 65532 65532  4096 Aug 26 10:36 dashboards
   drwxrwsr-x 8 root  65532  4096 Aug 26 10:34 data
   drwx--S--- 2 65532 65532  4096 Aug 26 10:36 log
   drwxrws--- 2 root  65532 16384 Aug 26 10:34 lost+found
   drwx--S--- 2 65532 65532  4096 Aug 26 10:36 rules
   drwx--S--- 2 65532 65532  4096 Aug 26 10:36 tmp
   ```

   b. After copying, go from the mount point to the / or /home directory to avoid blocking the subsequent unmounting of the volume.

7. Delete the temporary task, pod, PVC, and volume of the Metrics service:

   a. Delete the temporary task with the following command:

   ```
   sudo k0s kubectl delete job tmp-metrics-job -n kuma
   ```

   b. Make sure that the metrics-<UID> pod of the temporary task is not present in the output of the following command:

   ```
   sudo k0s kubectl get pods -n kuma
   ```

   c. If the pod does not disappear within a few minutes, force its deletion. The pod name must be taken from the output of the previous command.

   ```
   sudo k0s kubectl delete pod tmp-metrics-job-<UID> -n kuma --force
   ```

   d. Make sure that the Metrics service volume is in the detached state.

   ```
   sudo k0s kubectl get volume <volume_name> -n longhorn-system -o json | jq
   '.status.state'
   ```

8. Start the Metrics service.

   a. Run the following command to start the Metrics service with the number of replicas increased to 1.

   ```
   sudo k0s kubectl scale deployment/metrics -n kuma --replicas=1
   ```

   b. Make sure that the Metrics service is running and that the metrics are available in the KUMA web interface.

The Metrics service volume is successfully shrunk without data loss.

Shrinking a Metrics service volume with data loss

To shrink a Metrics service volume with data loss:

1. Get the name of the volume used by the Metrics service. You will need this volume name to locate volume data and volume replicas. Replica names begin with the volume name.

   ```
   sudo k0s kubectl get pvc metrics -n kuma -o json | jq '.spec.volumeName'
   ```

2. Stop the Metrics service.

```
sudo k0s kubectl scale deployment/metrics -n kuma --replicas=0
```

3. Make sure that the metrics-<UID> pod of the Metrics service is not present in the command output.

```
sudo k0s kubectl get pods -n kuma
```

4. If the pod does not disappear within a few minutes, force its deletion. The pod name must be taken from the output of the previous command.

```
sudo k0s kubectl delete pod metrics-<UID> -n kuma --force
```

5. Make sure that the Metrics service volume is in the detached state.

```
sudo k0s kubectl get volume <volume_name> -n longhorn-system -o json | jq
'.status.state'
```

6. Delete the PVC of the Metrics service.

```
sudo k0s kubectl delete pvc metrics -n kuma
```

7. Make sure the Metrics service volume is not present in the output of the following command:

```
sudo k0s kubectl get volume <volume_name> -n longhorn-system
```

On all worker nodes, directories with names beginning with the name of the deleted volume should disappear from the /opt/longhorn/replicas directory. If this does not happen, you need to delete them manually. Take care not to delete replicas that belong to other volumes by mistake.

8. Specify the value in the 'storage' line, then create a PVC for the Metrics service.

```
sudo k0s kubectl apply -f - <<EOF
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: metrics
  namespace: kuma
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: kuma-network-storage
  resources:
    requests:
      storage: 30Gi
EOF
```

9. Start the Metrics service, increasing the number of replicas to 1.

```
sudo k0s kubectl scale deployment/metrics -n kuma --replicas=1
```

10. Make sure that the Metrics service is running and that the metrics are available in the KUMA web interface.

The Metrics service volume is successfully shrunk with data loss.

Migrating alert events to ClickHouse after upgrading KUMA from version 3.4.x

After upgrading KUMA from version 3.4.x to version 4.0, you need to migrate alert events from MongoDB to ClickHouse and specify a storage in the alert filling settings.

After upgrading KUMA, the installer stops the kuma-mongodb service, which you must start manually before migrating alert events. If the KUMA Core is deployed in a Kubernetes cluster, the MongoDB container is also not started after the upgrade, so you need to start it before the migration.

If you need to abort the alert event migration command, press **Ctrl+C** in the terminal. A notification appears stating that the command will be stopped. The current batch of events is safely transferred to the storage, the migration progress is retained, and the process can be continued the next time you run the command.

Alert event migration is available for the following installation types:

- KUMA in a non-high-availability configuration
- Distributed installation with KUMA Core in a Kubernetes cluster

Migrating alert events when upgrading KUMA in a non-high-availability configuration

To migrate alert events in a non-high-availability configuration:

1. In the KUMA web interface, in the **Resources** → **Active services** section, select the storage to which you want to move the alert events and click **Copy cluster ID**. You will need the ID of the storage cluster when running the alert event migration command.

2. On the KUMA Core server, start kuma-mongodb.service using the following command:

   ```
   sudo systemctl start kuma-mongodb.service
   ```

3. Go to the /opt/kaspersky/kuma directory and run the following command, substituting the values into the options and substituting the previously copied storage cluster ID into the `--cluster-id` option:

   ```
   ./kuma tools migrate --mongo="mongodb://<name of the MongoDB host>:<port>" --core-url="https://<FQDN of the KUMA Core server>:<port>" --core-dir="<KUMA Core data storage directory: depending on the installation type, /opt/kaspersky/kuma/core/00000000-0000-0000-0000-000000000000 or /opt/kaspersky/kuma/core>" --cluster-id="<ID of the storage cluster where you want to move KUMA 3.4 alerts>" --batch=<number of alerts per batch> 2>&1 | tee /tmp/migrate.log | grep -v "debug"
   ```

   Example:

   ```
   ./kuma tools migrate --mongo="mongodb://localhost:27017" --core-url="https://kuma.example.com:7210" --core-dir="/opt/kaspersky/kuma/core/00000000-0000-0000-0000-000000000000" --cluster-id="0a123456-789a-0123-ab12-a3fab45ab6a7" --batch=10000 2>&1 | tee /tmp/migrate.log | grep -v "debug"
   ```

   Possible settings for alert event migration from MongoDB to ClickHouse

   The complete output of the command is saved to the /tmp/migrate.log file.

Migration can take a long time if there are many alerts. To speed up the migration process, you can use the `--batch` option and pre-allocate additional disk space and RAM. For example, you can migrate 150,000 alerts in one iteration lasting approximately three minutes with 12 GB of RAM for the migrator and MongoDB and 30 GB of disk space with the `--batch` option set to `150000`. In this example, the values and migration duration are given as a rough estimate. The duration may differ depending on the number of alerts and the number of events in alerts, the available disk space and RAM, and the number of alert events per batch specified in the `--batch` option. You can adapt these values to your needs.

You can also speed up the migration if you want to migrate only some alert events, for example, only alert events for the last year. In this case, you can specify the `--time-limit`, which corresponds to the **Alert retention period, days** setting.

For example, you have alert events for the past year and a half, and you want to migrate alert events for the last year. If you initiate the migration on August 18, 2025, and the **Alert retention period, days** is set to **365**, then set the `--time-limit` option to August 19, 2024 as follows: `--time-limit="Sun, 19 Aug 2024 00:01:00 UTC"`.

Only alert events for the last 365 days are migrated. This speeds up migration and avoids moving unwanted events.

If you do not specify the `--time-limit`, all alert events are migrated to ClickHouse, but this may take more time, and alert events older than the specified alert retention period will be subsequently deleted from ClickHouse.

4. After running the command, stop the mongodb service and disable it with the following commands:

   ```
   sudo systemctl stop kuma-mongodb.service
   ```

   ```
   sudo systemctl disable kuma-mongodb.service
   ```

Alert events are migrated.

After completing the KUMA upgrade and migrating alert events, the storage for new alert events is not automatically specified. The storage cluster you have specified is only for the migration of existing alert events. For new alerts to be filled with events, you must manually specify the **Storage** in the alert filling settings. Until you specify this setting, new alerts will be generated without events.

Migrating alert events after upgrading an installation with KUMA Core in a Kubernetes cluster

To migrate alert events in a KUMA installation with KUMA Core in a Kubernetes cluster:

1. In the KUMA web interface, in the **Resources** → **Active services** section, select the storage to which you want to move the alert events and click **Copy ID**. You will need the ID of the storage cluster when running the alert event migration command.

2. Start the MongoDB container in the KUMA Core pod by running the following command on the master controller:

   ```
   sudo k0s kubectl apply -f /root/k0s/core-manifest.yaml
   ```

   The KUMA Core is restarted.

3. Connect to the master controller of the k0s cluster and start the migration of alert events by substituting the previously obtained storage cluster identifier in the `--cluster-id` option:

```
k0s kubectl exec -it deployment/core-deployment -c core -n kuma -- kuma tools migrate
--mongo="mongodb://127.0.0.1:27017" --core-url="https://core:7210" --core-
dir="/opt/kaspersky/kuma/core" --cluster-id="ec9e34bc-8c05-4f95-bb37-cfd5fe4b354a" --
wd="/opt/kaspersky/kuma/core" --batch=10000 2>&1 | tee /tmp/migrate.log | grep -v
"debug"
```

Possible settings for alert event migration from MongoDB to ClickHouse

The complete output of the command is saved to the /tmp/migrate.log file on the host where you run `k0s kubectl`, not on the volume of the pod.

Migration can take a long time if there are many alerts. If the SSH session is interrupted, the command continues to run on the pod. In this case, you do not need to run the command again. You can watch the execution of the command in the processes on the worker node where the KUMA Core is running:

```
sudo watch 'ps -faux | grep "kuma tools migrate"'
```

To speed up the migration process, you can use the `--batch` option and pre-allocate additional disk space and RAM. For example, you can migrate 150,000 alerts in one iteration lasting approximately three minutes with 12 GB of RAM for the migrator and MongoDB and 30 GB of disk space with the `--batch` option set to 150000. In this example, the values and migration duration are given as a rough estimate. The duration may differ depending on the number of alerts and the number of events in alerts, the available disk space and RAM, and the number of alert events per batch specified in the `--batch` option. You can adapt these values to your needs.

You can also speed up the migration if you want to migrate only some alert events, for example, only alert events for the last year. In this case, you can specify the `--time-limit`, which corresponds to the **Alert retention period, days** setting.

For example, you have alert events for the past year and a half, and you want to migrate alert events for the last year. If you initiate the migration on August 18, 2025, and the **Alert retention period, days** is set to **365**, then set the `--time-limit` option to August 19, 2024 as follows: `--time-limit="Sun, 19 Aug 2024 00:01:00 UTC"`.

Only alert events for the last 365 days are migrated. This speeds up migration and avoids moving unwanted events.

If you do not specify the `--time-limit`, all alert events are migrated to ClickHouse, but this may take more time, and alert events older than the specified alert retention period will be subsequently deleted from ClickHouse.

4. After successfully migrating alert events, we recommend starting the KUMA Core without the MongoDB conainer. To do this, on the master controller, run the KUMA Core without the MongoDB container using the following command:

   ```
   sudo k0s kubectl apply -f /root/k0s/core-manifest-no-mongodb.yaml
   ```

   The KUMA Core is restarted.

Alert events are migrated.

After completing the KUMA upgrade and migrating alert events, the storage for new alert events is not automatically specified. The storage cluster you have specified is only for the migration of existing alert events. For new alerts to be filled with events, you must manually specify the **Storage** in the alert filling settings. Until you specify this setting, new alerts will be generated without events.

After migrating alert events to KUMA 4.0, you need to take into account the special considerations involved in their backup and restoration processes using the REST API.

Settings for alert event migration from MongoDB to ClickHouse

The following table lists and describes the available options for migrating alert events from MongoDB to ClickHouse.

Possible settings for alert event migration from MongoDB to ClickHouse

Setting	Type	Description	Default value
`--batch`	int	Size of the batch of events that is fully migrated to the storage in the course of a single iteration of the migration process. Increasing this value speeds up the migration, but requires more RAM and disk space. The value of this parameter must be determined based on the total number of alerts and available resources. To find out the total number of alerts, go to the **Alerts** section and click in the **Filters** drop-down list, click **Clear filters**. The total number of alerts is displayed in the **Found** field.	100
`--cluster-id`	string	ID of the cluster for storing alert events. You can get the cluster ID in the KUMA web interface in the **Active services** section. To do this, select the storage for the migration of alert events, right-click it to bring up the context menu, and select **Copy ID**.	–
`--core-dir`	string	The working directory of the KUMA Core with the service ID. The directory with certificates is located here.	–
`--core-url`	string	URL of the KUMA Core internal API. By default, the API is available on port 7210.	–
`--mongo`	string	URL of MongoDB to be connected to. Alert events that you want to migrate are stored here.	"mongodb://localhost:27017"
`--time-limit`	string	This option lets you migrate alert events created during the period from the specified timestamp (in RFC1123 format) up until the current moment. We recommend taking into account the **Alert retention period, days** because alert events older than the alert retention period are deleted from ClickHouse after migration. Example: You have alert events for the past year and a half, and you want to migrate only some alert events, for example, events for the last year. If you initiate the migration on August 18, 2025, and the **Alert retention period, days** is set to **365**, then set the `--time-limit` option to August 19, 2024 as follows: `--time-limit="Sun, 19 Aug 2024 00:01:00 UTC"`. As a result, only alert events for the last 365 days are migrated. This speeds up migration and avoids moving unwanted events. If you do not specify the `--time-limit`, all alert events are migrated to ClickHouse, but this may take more time, and alert events older than the specified alert retention period will be subsequently deleted from ClickHouse.	–
`--timeout`	int	ClickHouse storage connection timeout in seconds.	10
`--wd`	string	The working directory of the `migrate` command. Used for storing temporary and service files of the command. The process must have write permissions to this directory. Required option for an installation with KUMA Core in a Kubernetes cluster. If this option is not specified, the working directory is the current working directory from which the command is run.	–

Troubleshooting update errors

When updating KUMA, you may encounter the following errors:

- **Timeout error**

  When upgrading from version 2.0.x on systems that contain large amounts of data and are operating with limited resources, the system may return the `Wrong admin password` error message after you enter the administrator password. If you specify the correct password, KUMA may still return an error because KUMA could not start the KUMA Core service due to resource limit and a timeout error. If you enter the administrator password three times without waiting for the installation to complete, the update may end with a fatal error.

  To resolve the timeout error and complete the update successfully:

  1. Open a separate second terminal and run the following command to verify that the command output contains the timeout error line:

     ```
     journalctl -u kuma-core | grep 'start operation timed out'
     ```

     Timeout error message:

     `kuma-core.service: start operation timed out. Terminating.`

  2. After you find the timeout error message, in the `/usr/lib/systemd/system/kuma-core.service` file, change the value of the `TimeoutSec` parameter from 300 to 0 to remove the timeout limit and temporarily prevent the error from recurring.

  3. After modifying the service file, run the following commands in sequence:

     ```
     systemctl daemon-reload
     ```

     ```
     service kuma-core restart
     ```

  4. After running the commands and successfully starting the service in the second terminal, enter the administrator password again in your first terminal where the installer is prompting you for the password.

     KUMA will continue the installation. In resource-limited environments, installation may take up to an hour.

  5. After installation finishes successfully, in the `/usr/lib/systemd/system/kuma-core.service` file, set the `TimeoutSec` parameter back to 300.

  6. After modifying the service file, run the following commands in the second terminal:

     ```
     systemctl daemon-reload
     ```

     ```
     service kuma-core restart
     ```

  After you run these commands, the update will be succeed.

- **Invalid administrator password**

  The admin user password is required to automatically populate the storage settings during an update. If you entered an incorrect admin user password nine times during the `TASK [Prompt for admin password]`, the installer still performs the update, and the web interface is still available. However, the storage settings are not migrated, and the storages will show a red status.

  To fix the error and make the repositories available for use, update the storage settings:

  1. Go to the storage settings, manually fill in the fields of the ClickHouse cluster, and click **Save**.
  2. Restart the storage service.

     The storage service starts with the specified settings, and its status is green.

- **DB::Exception error**

  After updating KUMA, the storage may have a red status, and its logs may show errors.

  Example error:

  ```
  DB::Exception::Exception(std::__1::basic_string<char, std::__1::char_traits<char>,
  std::__1::allocator<char>> const&, int, bool) @ 0xda0553a in
  /opt/kaspersky/kuma/clickhouse/bin/clickhouse
  ```

  To restart ClickHouse, carry out the following command on the KUMA storage server:

  ```
  touch /opt/kaspersky/kuma/clickhouse/data/flags/force_restore_data && systemctl restart kuma-storage-<ID of the
  storage where the error was detected>
  ```

Fix the errors to successfully complete the update.

Reissuing k0s cluster certificates upon expiration

Problem

Controllers or worker nodes cannot connect; pods cannot be moved from one worker node to another.

Logs of the `k0scontroller` and `k0sworker` services contain multiple records with the following substring:

```
x509: certificate has expired or is not yet valid
```

Cause

Cluster service certificates are valid for 1 year from the time of creation. The k0s cluster used in the high-availability KUMA configuration automatically rotates all the service certificates it needs, but the rotation is performed only at startup of the `k0scontroller` service. If `k0scontroller` services on cluster controllers run without a restart for more than 1 year, service certificates become invalid.

How to fix

To fix the error, restart the `k0scontroller` *services one by one as root on each controller of the cluster.*

```
systemctl restart k0scontroller
```

This reissues the certificates.

To check the expiration dates of certificates on controllers, run the following commands as root:

```
find /var/lib/k0s/pki/ -type f -name "*.crt" -print|egrep -v 'ca.crt$'|xargs -L 1 -t  -i bash -c 'openssl x509  -noout -text -in {}|grep After'
```

```
find /var/lib/k0s/pki/etcd -type f -name "*.crt" -print|egrep -v 'ca.crt$'|xargs -L 1 -t  -i bash -c 'openssl x509 -noout -text -in {}|grep After'
```

You can find the names of certificate files and their expiration dates in the output of these commands.

Delete KUMA

To remove KUMA, use the Ansible tool and the inventory file that you have prepared.

To remove KUMA:

1. On the control machine, go to the installer directory:

   ```
   cd kuma-ansible-installer
   ```

2. Execute the following command:

   ```
   sudo ./uninstall.sh <inventory file>
   ```

   KUMA and all of its data is removed from the server.

The databases that were used by KUMA (for example, the ClickHouse storage database) and the data therein must be deleted separately.

Special considerations for removing a high availability configuration of KUMA

Which components need to be removed depends on the value of the `deploy_to_k8s` parameter in the inventory file that you are using to remove KUMA:

- If the setting is `true`, the Kubernetes cluster created during the KUMA installation is deleted.
- If the setting is `false`, all KUMA components except for the Core are removed from the Kubernetes cluster. The cluster is not deleted.

In addition to the KUMA components installed outside the cluster, the following directories and files are deleted on the cluster nodes:

- /usr/bin/k0s
- /etc/k0s/
- /var/lib/k0s/
- /usr/libexec/k0s/
- ~/k0s/ (for the ansible_user)
- /opt/longhorn/
- /opt/cni/
- /opt/containerd

When a cluster is being deleted, error messages may appear, however, it does not interrupt the installer.

- You can ignore such messages for the **Delete KUMA transfer job** and **Delete KUMA pod** tasks.
- For the **Reset k0s** task (if an error message contains the following text: "`To ensure a full reset, a node reboot is recommended.`") and the Delete k0s Directories and files task (if an error message contains the following text: "`I/O error: '/var/lib/k0s/kubelet/plugins/kubernetes.io/csi/driver.longhorn.io/`"), we recommend restarting the relevant host and trying to remove KUMA again with the same inventory file.

After removing KUMA, restart the hosts on which the KUMA or Kubernetes components were installed.

Working with tenants

Access to tenants is regulated in the settings of users. The *general administrator* has access to the data of all tenants. Only a user with this role can create and delete tenants.

Tenants are displayed in the table in the **Settings** → **Access** → **Tenants** section of the KUMA web interface. You can sort the table by clicking on 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.

To create a tenant:

1. In the KUMA web interface, in the **Settings** → **Access** → **Tenants**, click **Add**.

   The **Add tenant** window opens.

2. Specify the tenant name in the **Name** field. The name must contain 1 to 128 Unicode characters.
3. In the **EPS limit** field, specify the EPS quota for the tenant. This field is optional. The sum total of EPS quotas for all tenants must not exceed the EPS available under the license. However, KUMA does not check the values entered in this field.
4. If necessary, add a **Description** of the tenant. The description can contain no more than 256 Unicode characters.
5. Click **Create**.

The tenant is added. Press F5 to refresh the page. After refreshing the page, the created tenant is displayed in the web interface.

To delete a tenant:

1. In the **Settings** → **Access** → **Tenants** section of the KUMA web interface, select the relevant tenant by selecting the check box next to it, then in the toolbar, select **Delete**.
2. This opens the **Delete tenant?** window, which displays information about the tenant and prompts you to enter a code and confirm the deletion of the tenant. If you want to proceed with deleting the tenant, enter the code.
3. Click **OK**.

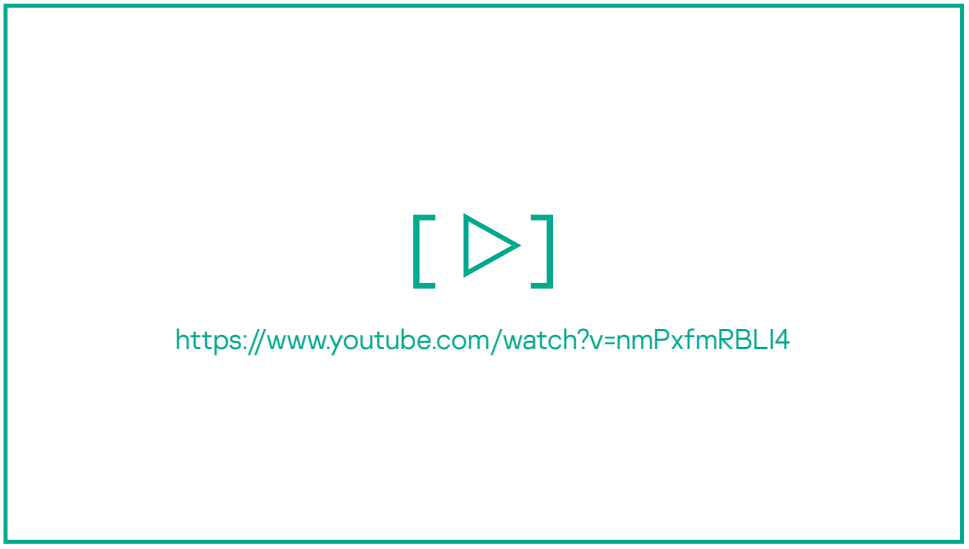
The tenant is deleted.

When you delete a tenant, the following happens:

- Services belonging to the tenant, with the exception of agents, are automatically stopped. In the **Resources** → **Active services** section, such services are displayed with a gray status.
- Reception and processing of events is terminated, and previously received events are deleted.
- All tenant resources are deleted.

- User accounts that have access only to the remote tenant are disabled—active sessions are terminated, and an error message is displayed when attempting to log in again.
- In the **Settings** → **Access** → **Domain authorization** section, under **Administration groups**, the role filters created for this tenant are deleted.

Agent services are not stopped automatically. If necessary, you can stop the Windows agent services manually in the **Start** → **Services** section; to stop the Linux agent services, you can press **Ctrl**+**C** in the terminal in which the agent was started.



https://www.youtube.com/watch?v=nmPxfmRBLI4

Selecting a tenant

If you have access to multiple tenants, KUMA lets you select which tenants' data will be displayed in the KUMA web interface.

To select a tenant for displaying data:

1. In the KUMA web interface, click **Selected tenants**.

   The tenant selection area opens.

2. Select the check boxes next to the tenants whose data you want to see in sections of the KUMA web interface.
3. You must select at least one tenant. You can use the **Search** field to search for tenants.
4. Click the tenant selection area by clicking **Selected tenants**.

Sections of the KUMA web interface will display only the data and analytics related to the selected tenants.

Your selection of tenants for data display will determine which tenants can be specified when creating resources, services, layouts, report templates, widgets, incidents, assets, and other KUMA settings that let you select a tenant.

Tenant affiliation rules

Tenant inheritance rules

It is important to track which tenant owns specific objects created in KUMA because this determines who will have access to the objects and whether or not interaction with specific objects can be configured. Tenant identification rules:

- The tenant of an object (such as a service or resource) is determined by the user when the object is created.

  After the object is created, the tenant selected for that object cannot be changed. However, resources can be exported then imported into another tenant.

- The tenant of an alert and correlation event is inherited from the correlator that created them.

  The tenant name is indicated in the `TenantId` event field.

- If events of different tenants that are processed by the same correlator are not merged, the correlation events created by the correlator inherit the tenant of the event.

- The incident tenant is inherited from the alert.

Examples of multitenant interactions

Multitenancy in KUMA provides the capability to centrally investigate alerts and incidents that occur in different tenants. Below are some examples that illustrate which tenants own certain objects that are created.

When correlating events from different tenants in a common stream, **you should not** group events by tenant. In other words, the `TenantId` event field should not be specified in the **Identical fields** field in correlation rules. Events must be grouped by tenant only if you must not merge events from different tenants.

Services that must be accommodated by the capacities of the main tenant can be deployed only by a user with the general administrator role.

- **Correlation of events for one tenant, correlator is allocated for this tenant and deployed at the tenant**

  Condition:

  The collector and correlator are owned by tenant 2 (tenantID=2)

  Scenario:

  1. The collector of tenant 2 receives and forwards events to the correlator of tenant 2.
  2. When correlation rules are triggered, the correlator creates correlation events with tenantID=2.

3. The correlator forwards the correlation events to the storage partition for tenant 2.
4. An alert is created and linked to the tenant with tenantID=2.
5. The events that triggered the alert are appended to the alert.

An incident is created manually by the user. The incident tenant is determined by the tenant of the user. An alert is linked to an incident either manually or automatically.

- **Correlation of events for one tenant, correlator is allocated for this tenant and deployed at the main tenant**

Condition:

- The collector is deployed at tenant 2 and is owned by this tenant (tenantID=2).
- The correlator is deployed at the main tenant.

  The owner of the correlator is determined by the general administrator depending on who will investigate incidents of tenant 2: employees of the main tenant or employees of tenant 2. The owner of the alert and incident depends on the owner of the correlator.

Scenario 1. The correlator belongs to tenant 2 (tenantID=2):

1. The collector of tenant 2 receives and forwards events to the correlator.
2. When correlation rules are triggered, the correlator creates correlation events with tenantID=2.
3. The correlator forwards the correlation events to the storage partition of tenant 2.
4. An alert is created and linked to the tenant with tenantID=2.
5. The events that triggered the alert are appended to the alert.

Result 1:

- The created alert and its linked events can be accessed by employees of tenant 2.

Scenario 2. The correlator belongs to the main tenant (tenantID=1):

1. The collector of tenant 2 receives and forwards events to the correlator.
2. When correlation rules are triggered, the correlator creates correlation events with tenantID=1.
3. The correlator forwards the correlation events to the storage partition of the main tenant.
4. An alert is created and linked to the tenant with tenantID=1.
5. The events that triggered the alert are appended to the alert.

Result 2:

- The alert and its linked events cannot be accessed by employees of tenant 2.
- The alert and its linked events can be accessed by employees of the main tenant.

- **Centralized correlation of events received from different tenants**

  Condition:

  - Two collectors are deployed: one at tenant 2 and one at tenant 3. Both collectors forward events to the same correlator.
  - The correlator is owned by the main tenant. A correlation rule waits for events from both tenants.

  Scenario:

  1. The collector of tenant 2 receives and forwards events to the correlator of the main tenant.
  2. The collector of tenant 3 receives and forwards events to the correlator of the main tenant.
  3. When a correlation rule is triggered, the correlator creates correlation events with tenantID=1.
  4. The correlator forwards the correlation events to the storage partition of the main tenant.
  5. An alert is created and linked to the main tenant with tenantID=1.
  6. The events that triggered the alert are appended to the alert.

  Result:

  - The alert and its linked events cannot be accessed by employees of tenant 2.
  - The alert and its linked events cannot be accessed by employees of tenant 3.
  - The alert and its linked events can be accessed by employees of the main tenant.

- **The tenant correlates its own events, but the main tenant additionally provides centralized correlation of events.**

  Condition:

  - Two collectors are deployed: one on the main tenant and one on tenant 2.
  - Two correlators are deployed:
    - Correlator 1 is owned by the main tenant and receives events from the collector of the main tenant and correlator 2.
    - Correlator 2 is owned by tenant 2 and receives events from the collector of tenant 2.

  Scenario:

  1. The collector of tenant 2 receives and forwards events to correlator 2.
  2. When a correlation rule is triggered, the correlator of tenant 2 creates correlation events with tenantID=2.
     - Correlator 2 forwards the correlation events to the storage partition of tenant 2.
     - Alert 1 is created and linked to the tenant with tenantID=2.

- The events that triggered the alert are appended to the alert.
- Correlation events from the correlator of tenant 2 are forwarded to correlator 1.

3. The collector of the main tenant receives and forwards events to correlator 1.
4. Correlator 1 processes events of both tenants. When a correlation rule is triggered, correlation events with tenantID=1 are created.
   - Correlator 1 forwards the correlation events to the storage partition of the main tenant.
   - Alert 2 is created and linked to the tenant with tenantID=1.
   - The events that triggered the alert are appended to the alert.

Result:

- Alert 2 and its linked events cannot be accessed by employees of tenant 2.
- Alert 2 and its linked events can be accessed by employees of the main tenant.

- **One correlator for two tenants**

  If you do not want events from different tenants to be merged during correlation, you should specify the `TenantId` event field in the **Identical fields** field in correlation rules. In this case, the alert inherits the tenant from the correlator.

  Condition:

  - Two collectors are deployed: one at tenant 2 and one at tenant 3.
  - One correlator owned by the main tenant (tenantID=1) is deployed. It receives events from both tenants but processes them irrespective of each other.

  Scenario:

  1. The collector of tenant 2 receives and forwards events to the correlator.
  2. The collector of tenant 3 receives and forwards events to the correlator.
  3. When a correlation rule is triggered, the correlator creates correlation events with tenantID=1.
     - The correlator forwards the correlation events to the storage partition of the main tenant.
     - An alert is created and linked to the main tenant with tenantID=1.
     - The events that triggered the alert are appended to the alert.

  Result:

  - Alerts that were created based on events from tenants 2 and 3 are not available to employees of these tenants.
  - Alerts and their linked events can be accessed by employees of the main tenant.

Managing users

It is possible for multiple users to have access to KUMA. Users are assigned user roles, which affect the tasks the users can perform. The same user may have different roles with different tenants. However, you cannot assign roles to yourself, even if your user account has the General administrator role; the displayed list of roles cannot be edited.

You can create or edit user accounts in the **Settings** → **Users** section of the KUMA web interface. Users are also created automatically in the solution if KUMA integration with Active Directory is enabled and the user is logging in to the KUMA web interface for the first time using their domain account.

All internal users whose accounts were created in the KUMA web interface, except for the admin account, must change their password when they first log into the KUMA web interface. Subsequently, these users must regularly change their password in accordance with corporate policies, but at least once every three months.

User accounts are displayed in the **Users** window of the KUMA web interface on the **Active users** and **Inactive users** tabs, depending on the user account status, *Active* or *Inactive*. You can use the **Search** field to look for users. You can sort the tables by clicking the column header and selecting **Ascending** or **Descending**.

You can create, edit, or disable user accounts. When editing user accounts (your own or the accounts of others), you can generate an API token for them.

To disable a user account:

In the **Settings** → **Access** → **Users** → **Active users** section of the KUMA web interface, select the check box next to the relevant user and click **Disable**.

User roles

A user's access to KUMA functionality depends on the user's role. A role is a set of rights and permissions to manage the solution. KUMA uses predefined roles with certain sets of access rights, as well as custom roles.

Predefined roles

When installing KUMA, the following predefined roles are created by default:

- *General administrator*—this role is designed for users who are responsible for the functioning of KUMA components. For example, they install KUMA components, monitor the performance of services within these components, create backups, and add users. These users have full access to KUMA.
- *Tenant administrator*—this role is for users responsible for the core functionality of KUMA components owned by specific tenants.
- *Tier 2 analyst*—this role is for users responsible for configuring the KUMA system to receive and process events of a specific tenant. They also create and tweak correlation rules.

- *Tier 1 analyst*—this role is for users responsible for configuring the KUMA system to receive and process events of a specific tenant. They also create and tweak correlation rules. Users with this role have fewer privileges than Tier 2 analysts.
- *Junior analyst*—this role is for users dealing with immediate security threats of a specific tenant. A user with this role can check resources of the Shared tenant through the REST API.
- *Interaction with NCIRCC*—this role can be selected if the license includes the GosSOPKA module. Users with this role receive notifications by default.
- *Access to CII*—this role can be selected if the license includes the GosSOPKA module. Users with this role receive notifications by default.
- *Read shared resources*—this role is intended for managing the Shared tenant. Users with this role have read access to shared resources.
- *Manage shared resources*—this role is intended for managing the Shared tenant. Users with this role have edit access to shared resources.

> The Interaction with NCIRCC and Interaction with CII roles are displayed only if your license covers the GosSOPKA module.

Each predefined role has the rights to perform certain actions in KUMA.

Custom roles

If existing predefined roles are inadequate for a user, you can create and use your own unique sets of rights and permissions by setting up custom roles.

In the **Settings** → **Access** → **Roles** section, you can perform the following actions:

- Creating custom roles, including by cloning existing roles (including predefined roles)
- Managing the settings of custom roles
- Viewing the set of rights and permissions of a role, the list of users with the selected role, as well as tenants of user roles
- Deleting custom roles

Rights of predefined roles

In KUMA, permissions are assigned to predefined user roles with regard to sections of the solution. Information about user permissions in various sections is given in the tables below.

User permissions for the Reports section

The Reports section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
Create report template	yes	yes	yes	yes	no	no	no	no	no
View and edit templates and reports	yes	yes	yes	yes	no	no	no	no	no
Generate reports	yes	yes	yes	yes	no	no	no	no	no
View generated reports	yes	yes	yes	yes	yes	no	no	no	no
Modify generated reports	yes	yes	yes	yes	no	no	no	no	no
Export generated reports	yes	yes	yes	yes	yes	no	no	no	no
Delete templates and generated reports	yes	yes	yes	yes	no	no	no	no	no
Manage report generation settings	yes	yes	yes	yes	no	no	no	no	no
Duplicate report template	yes	yes	yes	yes	no	no	no	no	no
Open the generated report by email	yes	yes	yes	yes	yes	no	no	no	no

User permissions for the Dashboard section

The Dashboard section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View data on the dashboard and change layouts	yes	yes	yes	yes	yes	no	no	yes	yes
View the Universal layout	yes	yes	yes	yes	yes	no	no	yes	yes
Add layouts	yes	yes	yes	yes	no	no	no	no	no
Edit and rename layouts	yes	yes	yes	yes	no	no	no	no	no
Delete layouts	yes	yes	yes	yes	no	no	no	no	no
Enable and disable the TV mode	yes	yes	yes	yes	yes	no	no	yes	yes

User permissions for the Resources section

The Resources → Services & Resources → Services → Active services section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the list of active services	yes	yes	yes	yes	yes	no	no	yes	yes
View and delete partitions in storage	yes	no	no	no	no	no	no	no	no
Download service logs	yes	yes	yes	yes	yes	no	no	yes	yes
Copy the service ID	yes	yes	yes	yes	yes	no	no	yes	yes
Refresh the table of active services	yes	yes	yes	yes	yes	no	no	yes	yes
Go to Events	yes	yes	yes	yes	yes	no	no	yes	yes
View the contents of the active list	yes	yes	yes	yes	no	no	no	no	no
View the content of a context table	yes	yes	yes	yes	no	no	no	no	no
Import/export/clear/add/edit/delete active list content	yes	yes	yes	yes	no	no	no	no	no
Import/export/clear/add/edit/delete context table content	yes	yes	yes	yes	no	no	no	no	no
Create a set of resources for services	yes	yes	yes	no	no	no	no	no	no
Create a service under Resources → Services → Active services	yes	yes	no	no	no	no	no	no	no
Delete services	yes	yes	no	no	no	no	no	no	no
View services that have the gray status	yes	no	no	no	no	no	no	no	no
Restart services	yes	yes	no	no	no	no	no	no	no
Update the settings of services	yes	yes	yes	no	no	no	no	no	no
Reset certificates	yes	yes	no	no	no	no	no	no	no

User permissions for the Data mining section

The Data mining section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the scheduler configuration	yes	yes	yes	no	no	no	yes	no	no
Create a scheduler	yes	yes	yes	no	no	no	no	no	no
Link storages and correlators	yes	yes	yes	no	no	no	no	no	no
Run	yes	yes	yes	no	no	yes	no	no	no

User permissions for the Resources configuration and Services configuration sections

The Resources → Resources configuration and Configuring services and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the list of resource and service configurations	yes	yes	yes	yes	no	yes	yes	no	no
Create a service configuration	yes	yes	yes	yes	no	no	no	no	no
Create a resource configuration	yes	yes	yes	yes	no	yes	no	no	no
Duplicate the configuration of a resource/service	yes	yes	yes	yes	no	yes	no	no	no
Edit the configuration of a resource/service	yes	yes	yes	yes	no	yes	no	no	no
Delete the configuration of a resource/service	yes	yes	yes	yes	no	yes	no	no	no
Import the configuration of a resource/service from a file	yes	yes	yes	yes	no	yes	no	no	no
View the repository, import the resources from the repository	yes	yes	yes	no	no	no	no	no	no
Export the configuration of a resource/service	yes	yes	yes	yes	no	yes	yes	no	no
Export correlation rules to MITRE ATT&CK Navigator	yes	yes	yes	yes	no	no	no	no	no
Create and edit a resource configuration in the Shared tenant	yes	no	no	no	no	yes	yes	no	no
View a resource in the Shared tenant (including linking, export, duplication)	yes	no	no	no	no	yes	yes	no	no

User permissions for the List of event sources section

The Source status → List of event sources section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View sources of events	yes	yes	yes	yes	yes	yes	no	yes	yes
Change sources of events	yes	yes	yes	no	no	yes	no	no	no
Delete sources of events	yes	yes	yes	no	no	yes	no	no	no

User permissions for the Monitoring policies section

The Source status → Monitoring policies section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View monitoring policies	yes	yes	yes	yes	yes	yes	yes	yes	yes
Create monitoring policies	yes	yes	yes	no	no	yes	no	no	no
Edit monitoring policies	yes	yes	yes	no	no	yes	no	no	no
Delete monitoring policies	yes	yes	yes	no	no	yes	no	no	no

User permissions for the Assets section

The Assets section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
Assets									
View the list of assets	yes	yes	yes	yes	yes	no	no	yes	yes
View the list of categories	yes	yes	yes	yes	yes	yes	yes	yes	yes
Add/edit/delete asset categories	yes	yes	yes	yes	no	no	no	no	no
Add asset categories in the Shared tenant	yes	no	no	no	no	no	no	no	no
Link assets to an asset category of the Shared tenant	yes	yes	yes	yes	no	yes	yes	no	no
Add assets	yes	yes	yes	yes	no	no	no	no	no
Edit assets	yes	yes	yes	yes	no	no	no	no	no
Delete assets	yes	yes	yes	yes	no	no	no	no	no
Import assets from Kaspersky Security Center	yes	yes	yes	yes	no	no	no	no	no
Start tasks on assets in Kaspersky Security Center	yes	yes	yes	yes	no	no	no	no	no
Run tasks to move an asset to a Kaspersky Security Center administration group	yes	yes	yes	yes	no	no	no	no	no
Run tasks on assets in Kaspersky Endpoint Detection and Response	yes	yes	yes	yes	no	no	no	no	no
Initiating the change of asset status in KICS for Networks	yes	yes	yes	yes	no	no	no	no	no
Confirm updates to fix the asset vulnerabilities and accept the licensing agreements	yes	yes	no	no	no	no	no	no	no

Editing CII categorization in the asset card	yes	no	no	no	no	no	no	no	yes
Editing custom fields of assets (**Settings** → **Other** → **Assets**)	yes	yes	yes	yes	no	no	no	no	no

User permissions for the Alerts section

The Alerts section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the list of alerts	yes	yes	yes	yes	yes	no	no	yes	yes
Change the severity of alerts	yes	yes	yes	yes	yes	no	no	yes	yes
Open the details of alerts	yes	yes	yes	yes	yes	no	no	yes	yes
Assign responsible users	yes	yes	yes	yes	yes	no	no	yes	yes
Close alerts	yes	yes	yes	yes	yes	no	no	yes	yes
Add comments to alerts	yes	yes	yes	yes	yes	no	no	yes	yes
Attach an event to alerts	yes	yes	yes	yes	yes	no	no	yes	yes
Detach an event from alerts	yes	yes	yes	yes	yes	no	no	yes	yes
Edit and delete someone else's filters	yes	yes	no	no	no	no	no	no	no

User permissions for the Incidents section

The Incidents section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the list of incidents	yes	yes	yes	yes	yes	no	no	yes	yes
Create blank incidents	yes	yes	yes	yes	yes	no	no	yes	yes
Manually create incidents from alerts	yes	yes	yes	yes	yes	no	no	yes	yes
Change the severity of incidents	yes	yes	yes	yes	yes	no	no	yes	yes
Open the incident details	yes	yes	yes	yes	yes	no	no	yes	yes
Assign executors	yes	yes	yes	yes	yes	no	no	yes	yes
Close incidents	yes	yes	yes	yes	yes	no	no	yes	yes
Add comments to incidents	yes	yes	yes	yes	yes	no	no	yes	yes
Attach alerts to incidents	yes	yes	yes	yes	yes	no	no	yes	yes

Detach alerts from incidents	yes	yes	yes	yes	yes	no	no	yes	yes
Edit and delete someone else's filters	yes	yes	no	no	no	no	no	no	no
Export incidents to RuCERT	yes	no	no	no	no	no	no	yes	no
Send files to NCIRCC	yes	no	no	no	no	no	no	yes	no
Download files sent to NCIRCC	yes	no	no	no	no	no	no	yes	no
Export additional incident data to NCIRCC upon request	yes	no	no	no	no	no	no	yes	no
Send messages to NCIRCC	yes	no	no	no	no	no	no	yes	no
View messages from NCIRCC	yes	no	no	no	no	no	no	yes	no
View incident data exported to NCIRCC	yes	no	no	no	no	no	no	yes	no

User permissions for the Events section

The Events section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the list of events	yes	yes	yes	yes	yes	no	no	yes	yes
Search events	yes	yes	yes	yes	yes	no	no	yes	yes
Open the details of events	yes	yes	yes	yes	yes	no	no	yes	yes
Open statistics	yes	yes	yes	yes	yes	no	no	yes	yes
Perform a retrospective scan	yes	yes	yes	no	no	no	no	no	no
Export events to a TSV file	yes	yes	yes	yes	yes	no	no	yes	yes
View filters (search query)	yes	yes	yes	yes	yes	yes	yes	yes	yes
Add (save) filter	yes	yes	yes	yes	yes	yes	no	yes	yes
Edit or delete someone else's filter folders	yes	yes	no	no	no	no	no	no	no
Delete own filters	yes	yes	yes	yes	yes	yes	no	yes	yes
Delete other users' filters	yes	yes	no	no	no	no	no	no	no
Start ktl enrichment	yes	yes	yes	yes	no	no	no	no	no
Run tasks on Kaspersky Endpoint Detection and Response assets in event details	yes	yes	yes	yes	no	no	no	no	no
Create presets	yes	yes	yes	yes	yes	yes	no	yes	yes
Delete presets	yes	yes	yes	yes	yes	yes	no	yes	yes
View and use presets	yes	yes	yes	yes	yes	yes	yes	yes	yes
Run queries in the background	yes	yes	yes	yes	yes	no	no	yes	yes

User permissions for the Users and actions section

The Settings → Access → Users section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the list of users	yes	no	no	no	no	no	no	no	no
Add a user	yes	no	no	no	no	no	no	no	no
Edit a user	yes	no	no	no	no	no	no	no	no
Generate token	yes	yes	yes	yes	yes	yes	yes	yes	yes
Change access rights for a token	yes	yes	yes	yes	yes	yes	yes	yes	yes
View the data of their own profile	yes	yes	yes	yes	yes	yes	yes	yes	yes
Edit the data of their own profile	yes	yes	yes	yes	yes	yes	yes	yes	yes

User permissions for the LDAP server section

The Settings → Integrations → LDAP server section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the LDAP connection settings	yes	yes	yes	yes	no	no	no	no	no
Edit the LDAP connection settings	yes	yes	no	no	no	no	no	no	no
Delete the configuration of an entire tenant from the settings	yes	yes	no	no	no	no	no	no	no
Import assets	yes	yes	no	no	no	no	no	no	no

User permissions for the Tenants section

The Settings → Access → Tenants section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the list of tenants	yes	no	no	no	no	no	no	no	no
Add tenants	yes	no	no	no	no	no	no	no	no
Change tenants	yes	no	no	no	no	no	no	no	no
Export tenants	yes	no	no	no	no	no	no	no	no

This section is available only to the general administrator.

User permissions for the "Domain authentication" section

The "Settings → Access → Domain authentication" section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the Active Directory connection settings	yes	no	no	no	no	no	no	no	no
Edit the Active Directory connection settings	yes	no	no	no	no	no	no	no	no
Add filters based on roles for tenants	yes	no	no	no	no	no	no	no	no

This section is available only to the general administrator.

User permissions for the General section

The Settings → Other → General section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the SMTP connection settings	yes	no	no	no	no	no	no	no	no
Edit the SMTP connection settings	yes	no	no	no	no	no	no	no	no

This section is available only to the general administrator.

User permissions for the License section

The Settings → Other → License section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the list of added license keys	yes	no	no	no	no	no	no	no	no
Add license keys	yes	no	no	no	no	no	no	no	no
Delete license keys	yes	no	no	no	no	no	no	no	no

This section is available only to the general administrator.

User permissions for the Kaspersky Security Center section

The Settings → Integrations → Kaspersky Security Center section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the list of successfully integrated Kaspersky Security Center servers	yes	yes	yes	yes	no	no	no	no	no
Add Kaspersky Security Center connections	yes	yes	no	no	no	no	no	no	no
Delete Kaspersky Security Center connections	yes	yes	no	no	no	no	no	no	no
Delete the configuration of an entire tenant from the settings	yes	yes	no	no	no	no	no	no	no
Start the tasks for importing Kaspersky Security Center assets	yes	yes	no	no	no	no	no	no	no

User permissions for the KICS/KATA section

The Settings → Integrations → KICS/KATA section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View a list of KICS/KATA servers with which integration has been configured	yes	yes	no	no	no	no	no	no	no
Add and modify the settings of KICS/KATA integration	yes	yes	no	no	no	no	no	no	no
Delete the settings of KICS/KATA integration	yes	yes	no	no	no	no	no	no	no
Run the tasks to import assets from the KICS/KATA settings	yes	yes	no	no	no	no	no	no	no

User permissions for the Kaspersky Automated Security Awareness Platform section

The Settings → Integrations → Kaspersky Automated Security Awareness Platform section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the ASAP integration settings	yes	no	no	no	no	no	no	no	no
Edit the ASAP integration settings	yes	no	no	no	no	no	no	no	no

User permissions for the Kaspersky Endpoint Detection and Response section

The Settings → Integrations → Kaspersky Endpoint Detection and Response section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the connection settings	yes	yes	yes	yes	no	no	no	no	no
Add, edit and disconnect the connections when the distributed solution mode is enabled	yes	no	no	no	no	no	no	no	no
Enable the distributed solution mode	yes	no	no	no	no	no	no	no	no
Add connections when the distributed solution mode is disabled	yes	yes	no	no	no	no	no	no	no
Delete the connections when the distributed solution mode is disabled	yes	yes	no	no	no	no	no	no	no
Delete the configuration of an entire tenant from the settings	yes	yes	no	no	no	no	no	no	no

User permissions for the Kaspersky CyberTrace section

The Settings → Integrations → Kaspersky CyberTrace section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the CyberTrace integration settings	yes	no	no	no	no	no	no	no	no
Edit the CyberTrace integration settings	yes	no	no	no	no	no	no	no	no

This section is available only to the general administrator.

User permissions for the IRP / SOAR section

The Settings → Integrations → IRP / SOAR section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the settings for integration with IRP / SOAR	yes	no	no	no	no	no	no	no	no
Edit the IRP/SOAR integration settings	yes	no	no	no	no	no	no	no	no

This section is available only to the general administrator.

User permissions for the Kaspersky Threat Lookup section

The Settings → Integrations → Kaspersky Threat Lookup section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the Threat Lookup integration settings	yes	no	no	no	no	no	no	no	no
Edit the Threat Lookup integration settings	yes	no	no	no	no	no	no	no	no

This section is available only to the general administrator.

User permissions for the Alerts section

The Settings → Other → Alerts section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the parameters	yes	yes	yes	yes	no	no	no	no	no
Edit the parameters	yes	yes	yes	no	no	no	no	no	no
Delete the configuration of an entire tenant from the settings	yes	yes	yes	no	no	no	no	no	no

User permissions for the Automatic linking of alerts to incidents section

The Settings → Other → Incidents → Automatic linking of alerts to incidents section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the parameters	yes	yes	yes	yes	no	no	no	no	no
Edit the parameters	yes	no	no	no	no	no	no	no	no

This section is available for an account with the Tenant administrator, Tier 2 analyst, or Tier 1 analyst role if the role is assigned in the Main tenant.

User permissions for the Incident types section

The Settings → Other → Incidents → Incident types section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the categories reference	yes	yes	yes	yes	no	no	no	no	no
View the categories charts	yes	yes	yes	yes	no	no	no	no	no
Add categories	yes	yes	no	no	no	no	no	no	no

Edit categories	yes	yes	no	no	no	no	no	no	no
Delete categories	yes	yes	no	no	no	no	no	no	no

User permissions for the NCIRCC section

The Settings → Integrations → NCIRCC section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the parameters	yes	no	no	no	no	no	no	no	no
Edit the parameters	yes	no	no	no	no	no	no	no	no

User permissions for the Asset audit section

The Settings → Other → Assets → Asset audit section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
Create, clone and edit the settings	yes	yes	yes	no	no	no	no	no	no
View the parameters	yes	yes	yes	yes	no	no	no	no	no
Delete settings	yes	yes	yes	no	no	no	no	no	no

User permissions for the Repository update section

The Settings → Other → Repository update section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the parameters	yes	yes	yes	no	no	no	no	no	no
Edit the parameters	yes	no	no	no	no	no	no	no	no
Start the repository update task manually	yes	yes	yes	no	no	no	no	no	no

User permissions for the Assets section

The Settings → Other → Assets section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
Add, edit, and delete the asset fields	yes	no	no	no	no	no	no	no	no

User permissions for the Access to spaces section

The Settings → Access → Access to spaces section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
Viewing the list of sets	yes	no	no	no	no	no	no	no	no
Adding, editing, deleting a space set	yes	no	no	no	no	no	no	no	no
Setting the default space set	yes	no	no	no	no	no	no	no	no
Assigning a space set to a user	yes	no	no	no	no	no	no	no	no

User permissions for the Tags section

The Settings → Other → Tags section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View the list of tags	yes	yes	yes	yes	no	no	no	no	no

User permissions for the Extended event schema fields scheme

The Options → Other → Extended event schema fields section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
Viewing the table of extended event schema fields	yes	yes	yes	yes	yes	no	yes	no	no
Exporting extended event schema fields	yes	yes	yes	yes	yes	no	yes	no	no
Adding and modifying extended event schema fields	yes	yes	yes	yes	yes	no	no	no	no
Importing extended event schema fields	yes	yes	yes	yes	yes	no	no	no	no
Removing and disabling extended event schema fields	yes	no	no	no	no	no	no	no	no

User permissions for the Metrics section

The Metrics section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
Open metrics	yes	no	no	no	no	no	no	no	no

User permissions for the Task manager section

The Task manager section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
View a list of your own tasks	yes	yes	yes	yes	yes	no	no	yes	yes
Finish your own tasks	yes	yes	yes	yes	yes	no	no	yes	yes
Restart your own tasks	yes	yes	yes	yes	yes	no	no	yes	yes
View a list of all tasks	yes	yes	yes	yes	yes	no	no	yes	yes
Finish any task	yes	no	no	no	no	no	no	no	no
Restart any task	yes	yes	yes	yes	yes	no	no	yes	yes
Restart tasks of the Run query in the background type	yes	yes	yes	yes	yes	no	no	yes	yes
Stop tasks of the "Run query in the background" type, and initiate the execution of a query in the **Events** section	yes	yes	yes	yes	yes	no	no	yes	yes

User permissions for the CyberTrace section

The CyberTrace section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
Open the section	yes	no	no	no	no	no	no	no	no

This section is not displayed in the web interface unless CyberTrace integration is configured under **Settings** → **Integrations** → **Kaspersky CyberTrace**.

User permissions for the Access to tenant data section

The Access to tenant data section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources	Access to NCIRCC	Access to CII
Access to tenants	yes	yes	yes	yes	yes	no	no	yes	yes

Shared tenant	yes	yes	yes	yes	yes	yes	yes	yes	yes
Main tenant	yes	yes	yes	yes	yes	no	no	yes	yes

Creating a user role

When you create a user role, you can give this role permissions to create users and manage roles. Users with such a role can create another user account for themselves or a role with broad rights and permissions (including administrative control). We strongly recommend that you only assign administrator rights to trusted users.

To create a custom role:

1. In the **Settings** → **Access** → **Roles** section, click the **Add** button.

   A role is created in the enabled state, you can disable it in the table of roles or in the sidebar with the description of the role settings.

2. In the **Name** field, specify a name for the custom role containing 1 to 128 Unicode characters.

3. If necessary, add a description of the role (up to 128 characters).

4. Under **List of permissions**, select check boxes next to the required functional areas and select a permission in each area.

   The list of selectable functional areas includes individual functional areas and groups of functional areas. For example, the **KSC** group includes the following functional areas: **Common** (viewing the status of KSC tasks) and **Updates** (reading KSC updates). You can select a group of functional areas in its entirety or pick individual functional areas from a group.

   For the selected functional areas, you must specify the level of rights and permissions. The following permissions are available in KUMA:

   - **Manage Manage** grants full management permissions for the functional area (read, create, edit, delete).
   - **Manage created Manage created** grants full user permissions in the functional area (reading and duplicating all objects, modifying and deleting own objects) for objects created by the user.
   - **Read Read** grants read permissions.
   - **No rights** does not grant any permissions to manage objects.

   Only one permission level can be specified for each functional area.

   By default, when a role is created, the functional area is assigned the same permission level as the predefined Junior analyst role.

You can specify the permissions as follows:

- Specify the permissions separately for each selected functional area by selecting the corresponding rights in the **Permissions** column in the table of functional areas.
- Select all functional areas included in the group, and set a level of permissions for these by clicking buttons above the table.
- Select all groups of functional areas, and set a level of permissions level for these by clicking buttons above the table.

If you assign permissions using the buttons above the table, the permissions are assigned only to those functional areas to which they are applicable. For example, you can set **No rights** or **Read** permissions for the **Events** functional area. If you select the **Manage** level of permissions for the selected functional areas or groups of functional areas by clicking the relevant button above the table, the level of permissions is only changed for those functional areas to which **Manage** is applicable. For the **Events** functional area, for which this permission is not available, the permission is not changed.

AI-related functional areas are displayed in a role if the license covers the AI module, and functional areas related to NCIRCC and CII are displayed if the license covers the GosSOPKA module.

5. Click the **Create** button.

   After saving, the role appears in the list of roles that can be assigned to a user in the **Settings** → **Access** → **Users** section.

In KUMA 4.2, you cannot create a custom role with access to the Shared tenant and its resources. For users needing access to the Shared tenant, assign Shared tenant roles. By default, access to the Shared tenant is granted to the General Administrator and to roles that have the following permissions: Manage shared resources, Read shared resources, and Access to shared dashboards.

You can also create a custom role by copying an existing role (including a predefined role).

To create a role based on an existing role:

1. In the **Settings** → **Access** → **Roles** section, select a check box to pick the role from which you want to copy settings.
2. Click the **Duplicate** button above the table of roles.
3. In the sidebar that is displayed, if necessary, change the name, description, and permissions of the role.
4. Click the **Create** button.

Functional areas and permissions available to them

The following table lists KUMA functional area groups and the functional areas that they include. For each functional area, a description is given as well as a list of available authorization levels that can be set for custom roles.

Groups of functional areas and the functional areas in them

Functional area	Description	Possible permissions
Group of functional areas: Alerts		
Alerts	Working with alerts	• Manage • Read • No rights
Group of functional areas: Incidents		
Common	Working with incidents	• Manage • Read • No rights
Types	Managing incident types	• Manage • No rights
NCIRCC	Exporting incidents to RuCERT	• Manage • No rights
Group of functional areas: Events		
Events	Searching events	• Read • No rights
Group of functional areas: Response		
Adding an indicator to the list of indicators of compromise	Adding an indicator to the list of excluded CyberTrace indicators of compromise from the event card (manually)	• Manage • No rights
KATA	Response using KATA means	• Manage • No rights
KICS	Response using KICS means	• Manage • No rights
KSC	Response using KSC means	• Manage • Read • No rights
Active Directory	Response using Active Directory means	• Manage • No rights
KASAP	Move between KASAP groups	• Manage • Read • No rights
Group of functional areas: Assets		
Common	Managing assets	• Manage • Read • No rights
Asset categorization	Managing automatic asset categorization rules	• Manage • Read • No rights
Categorize CII assets	Read and edit settings when categorizing a CII asset in the asset card	• Manage • No rights

Functional area	Description	Possible permissions
Group of functional areas: Resources		
Collectors	Managing collector configuration	• Manage • Manage created • Read • No rights
Correlators	Managing correlator configuration	• Manage • Manage created • Read • No rights
Storages	Managing storage configuration	• Manage • Manage created • Read • No rights
Agents	Managing agent configuration	• Manage • Manage created • Read • No rights
Event routers	Managing event router configuration	• Manage • Manage created • Read • No rights
Normalizers	Managing normalizer configuration	• Manage • Manage created • Read • No rights
Connectors	Managing connector configuration	• Manage • Manage created • Read • No rights
Destinations	Managing destination configuration	• Manage • Manage created • Read • No rights
Filters	Managing filter configuration	• Manage • Manage created • Read • No rights
Active list	Managing active list configuration	• Manage • Manage created • Read • No rights
Context tables	Managing the configuration of context tables	• Manage • Manage created • Read • No rights
Dictionaries	Managing dictionary configuration	• Manage • Manage created • Read • No rights
Proxies	Managing proxy server configuration	• Manage • Manage created

Functional area	Description	Possible permissions
		• Read • No rights
Secrets	Managing secret configuration	• Manage • Manage created • Read • No rights
Notification templates	Managing notification template configuration	• Manage • Manage created • Read • No rights
Aggregation rules	Managing aggregation rule configuration	• Manage • Manage created • Read • No rights
Correlation rules	Managing correlation rule configuration	• Manage • Manage created • Read • No rights
Enrichment rules	Managing enrichment rule configuration	• Manage • Manage created • Read • No rights
Response rules	Managing response rule configuration	• Manage • Manage created • Read • No rights
Data collection and analysis rules	Managing data collection and analysis rules	• Manage • Manage created • Read • No rights
Segmentation rules	Managing segmentation rule configuration	• Manage • Manage created • Read • No rights
Dashboards	Managing dashboards	• Manage • Manage created • Read • No rights
Report templates	Managing report templates	• Manage • Manage created • Read • Read created • No rights
Importing resources	Importing any resources, folders, and asset categories, including overwriting existing ones	• Manage • No rights
Exporting resources	Exporting any resources, folders, and asset categories	• Read • No rights
Import from repository	Importing resources from a repository	• Manage • No rights
Repository	Reading resource repository	• Read • No rights
Directories	Managing directories	• Manage • Manage

Functional area	Description	Possible permissions
		created • Read • No rights
Field sets	Managing user-created field sets	• Manage • Manage created • Read • No rights
Panels of secrets	Viewing the panel of secrets in the **Resources** section	• Read • No rights
Group of functional areas: Creating exclusions from correlation rules		
Create exclusions	Creating exclusions from correlation rules	• Manage • No rights
Read and delete exclusions	Managing exclusions from correlation rules	• Manage • No rights
Group of functional areas: Active services		
Subgroup: Work with services		
Collector	Managing collector service	• Manage • Read • No rights
Correlator	Managing correlator service	• Manage • Read • No rights
Active lists on correlator	Managing the content of active list and context table on correlator	• Manage • Read • No rights
Storage	Managing storage service	• Manage • Read • No rights
Storage partitions	Managing storage partitions in active services	• Manage • No rights
Active queries to ClickHouse	Managing active queries to ClickHouse	• Manage • Read • No rights
Agent	Managing agent service	• Manage • Read • No rights
Core	Managing core service	• Manage • Read • No rights
Event router	Managing event router service	• Manage • Read • No rights
Subgroup: Actions		
Updating configuration	Updating active service configuration	• Manage • No rights
Restart	Restart an active service	• Manage • No rights
Reset certificate	Resetting active service certificate	• Manage • No rights
Download active service artifacts	Download active service artifacts	• Manage • No rights
Remaining services from deleted tenants	Reading and deleting services left behind by deleted tenants	• Manage • No rights

Functional area	Description	Possible permissions
Group of functional areas: Data retrieval and analysis		
Retrieve and analyze data	Managing data retrieval and analysis	• Manage • Read • No rights

Functional area	Description	Possible permissions
Group of functional areas: Tasks		
Retroscan	Managing retroscan background task	• Manage • Manage created • Read • No rights
Exporting events	Managing event export background task	• Manage • Manage created • Read • No rights
KTL	Managing KTL query background task	• Manage • Manage created • Read • No rights
KATA	Managing KATA query background task	• Manage • Manage created • Read • No rights
Download assets	Managing KSC/KICS asset download background task	• Manage • Manage created • Read • No rights
Download user accounts	Managing the download Active Directory user accounts background task	• Manage • Manage created • Read • No rights
Update KUMA content	Managing the update KUMA content background task	• Manage • Manage created • Read • No rights
Clear history	Managing clear resource history background task	• Manage • Manage created • Read • No rights
Update monitoring policy files	Managing the update monitoring policy files background task	• Manage • Manage created • Read • No rights
Export event sources	Managing the export event sources to CSV format background task	• Manage • Manage created • Read • No rights
KIRA	Managing background task, viewing KIRA query results and deobfuscation	• Manage • Manage created • Read • No rights
Searching events in the background	Managing the background event search background task	• Manage • Manage created • Read • No rights
Backing up partitions	Managing partition backup background task	• Manage • No rights

Functional area	Description	Possible permissions
Restoring partitions from backup	Managing the restore partitions from backup background task	• Manage • No rights
Import partition backup	Managing import partitions backup background task	• Manage • No rights
Group of functional areas: Settings		
Subgroup: Access – Users		
Common	Managing users	• Manage • Manage created • Read • No rights
No tenant access	Managing users that do not have access to tenants	• Manage • No rights
Credentials	Managing the creation of a REST API access token, changing a password	• Manage • Manage created • No rights
Subgroup: Access – Roles		
Roles	Managing custom role settings	• Manage • Read • No rights
Common	Managing the assignment of roles to users (except for the General administrator)	• Manage • Read • Read created • No rights
General administrator	Managing the assignment of the General administrator role	• Manage • No rights
CyberTrace administrator	Managing the administrator role in the CyberTrace web console	• Manage • No rights
Subgroup: Access – Tenants		
Tenant registry	Managing the list of tenants	• Manage • Read
Settings	Managing KUMA-specific tenant settings (for example, EPS limits)	• Manage • No rights
Subgroup: Access – Active Directory		
Managing domain authentication settings	Managing ActiveDirectory, FreeIPA authorization settings	• Manage • Read • No rights
Accounts	LDAP server integration by tenant	• Manage • Read • No rights
Account details	Active Directory account details	• Read • No rights
Subgroup: Access – Security policies		
Security policies	Managing security policy configuration	• Manage • Read • No rights
Subgroup: Access – Access to spaces		
Actions with space sets	Managing event space sets	• Manage • No rights
Assigning space sets to users	Assigning and reading custom space sets; assigning default space sets	• Manage • No rights

Functional area	Description	Possible permissions
Subgroup: Integration		
KTL	Managing KTL integration settings	• Manage • Read • No rights
CyberTrace	Managing CyberTrace integration settings	• Manage • Read • No rights
KSC	Managing KSC updates	• Manage • Read • No rights
KIRA	Managing KIRA integration settings	• Manage • Read • No rights
KSN	Managing KSN integration settings	• Manage • Read • No rights
KICS	Managing KICS integration settings	• Manage • Read • No rights
KATA single	Managing KATA integration settings	• Manage • Read • No rights
KATA multi	Managing KATA integration settings (distributed solution)	• Manage • Read • No rights
NCIRCC	Managing NCIRCC settings	• Manage • No rights
KSN	Managing KSN integration settings	• Manage • Read • No rights
KASAP	Managing KASAP integration settings	• Manage • Read • No rights
KIRA	Managing KIRA integration settings	• Manage • Read • No rights
KSN	Managing KSN integration settings	• Manage • Read • No rights
Subgroup: Alerts		
Linking segmentation rules to alerts	Managing the linking of segmentation rules to alerts	• Manage • Read • No rights
Linking notification templates to alerts	Managing the linking of notification templates to alerts	• Manage • Read • No rights
Alerts	Managing alert creation settings and alert event storage	• Manage • Read • No rights
Subgroup: License		
License	Managing licensing	• Manage • Read • No rights
Subgroup: Incidents		
Incidents	Managing incident configuration	• Manage • Read • No rights

Functional area	Description	Possible permissions
Subgroup: KUMA repository update		
Updating the KUMA repository	Managing KUMA repository update settings	• Manage • Manage created • Read • No rights
Subgroup: Assets		
Importing assets	Managing asset import settings	• Manage • Read • No rights
Asset audit	Managing asset audit settings	• Manage • Read • No rights
KSC	Managing KSC asset import settings	• Manage • Read • No rights
Subgroup: Resource tag list		
Resource tag list	Managing KUMA resource tags	• Read • No rights
Subgroup: Service monitoring		
Service monitoring. Configuring thresholds	Managing thresholds for alerts of each type of service	• Manage • Read • No rights
Subgroup: Extended event schema fields		
Deleting and modifying KUMA extended event schema fields	Deleting and toggling the `disabled` status of KUMA extended event schema fields	• Manage • No rights
KUMA extended event schema fields	Managing KUMA extended event schema fields (except for deleting and toggling the `disabled` status)	• Manage • Read • No rights
Subgroup: Ignore TTL		
Ignore TTL	Managing the ignore TTL setting	• Manage • No rights
Subgroup: Event source discovery		
Event source discovery	Managing event source discovery settings	• Manage • Read • No rights
AI score and AI status of assets		
AI score and AI status of assets	Managing API configuration for `sequence appraiser` setting	• Manage • No rights
Subgroup: Rvision		
Rvision	Managing Rvision settings	• Manage • No rights
Subgroup: Access to REST API permissions settings		
Access to REST API permissions settings	Managing the list of available endpoints settings	• Manage • Manage created • No rights

Functional area	Description	Possible permissions
Group of functional areas: System operations		
Creating a backup copy	Creating a backup copy of KUMA Core	• Manage • No rights
Restoring	Restoring KUMA from a backup copy.	• Manage • No rights
KUMA web application certificates	Managing KUMA web application certificate	• Manage • No rights
KUMA internal certificates	Managing KUMA internal certificates	• Manage • No rights
Maintenance	Maintenance	• Manage • No rights
Group of functional areas: CyberTrace		
CyberTrace	CyberTrace Web Console access	• Read • No rights
Group of functional areas: Event sources		
Event sources	Managing event sources	• Manage • Read • No rights
Monitoring policies	Managing monitoring policies	• Manage • Read • No rights
Inline monitoring policies	Managing inline monitoring policies	• Manage • No rights
Group of functional areas: Predefined entities		
Predefined entities	Managing predefined entities	• Manage • Read • No rights
Group of functional areas: Notifications		
Receiving SMTP notifications	Receiving SMTP notifications	• Read • No rights

Editing user role settings

You cannot manage the settings of predefined system roles. You can duplicate a predefined role and view the cards of roles (the permissions of a role and the users to which the role is assigned).

To manage the settings of a custom role:

1. In the **Settings** → **Access** → **Roles** section, click the name of the custom role in the Name column of the table of roles.

2. On the displayed sidebar, enable or disable the toggle switch to activate or deactivate the role, if necessary.

   When you change the state of a role to make it inactive, the role becomes unavailable for assignment to users. If this role has been assigned to a user, the permissions granted by the role are stripped from the user. If this role was the only role assigned to the user, the user's session is terminated. In the **Settings** → **Access** → **Users** section, such a user is no longer displayed among active users, but is instead listed among inactive users. The card of such a user displays a warning about needing to assign at least one role to the user or to delete the user.

When you change the state of a role to make it active, the role is added to the list of roles that can be assigned to a user.

After activating a role, the user to which this role is assigned does not automatically become an active user in the **Settings** → **Access** → **Users** section. You need to activate the user.

By default, a custom role is created in an active state.

3. If necessary, edit the name, description, and permissions of the custom role. You can use the **Show functional areas** filter to select the authorization level that you need. KUMA displays only those functional areas to which your selected permissions have been assigned.

4. Click the **Save** button.

Viewing role settings

To view the set of permissions of a role:

In the **Settings** → **Access** → **Roles** section, click a role name in the **Name** column of the table of user roles.

In the displayed sidebar on the right, under **List of permissions**, a set of permissions by functional area is displayed for the role.

To view the list of users with the selected role and tenants in which the role will be assigned to users:

1. Go to the **Settings** → **Access** → **Roles** section.

2. Do one of the following:

   - Click the number indicated for the role in the **Users** column of the table of roles.

   - Click the name of the role in the **Name** column to open a window, and in that window, select the **Users and groups** tab.

   The opened window displays users to which the role has been assigned. You can view active and inactive users with their names, login names, email addresses. For each user, the number of assigned roles and the tenant of the user are also indicated. Clicking the link with the number specified for the role in the **Users** column lets you edit the user account.

Deleting a user role

To delete custom roles:

1. In the **Settings** → **Access** → **Roles** section, select check boxes to pick the role or roles that you want to delete.

2. Click the **Delete** button above the table of roles.

> The **Delete** button becomes clickable only when you select some custom roles for deletion. Predefined roles cannot be deleted.

3. To confirm deletion, do one of the following:

   - If the role has been assigned to a user, enter the confirmation code and click **OK**.
   - If the role has not been assigned to a user, click **OK**.

   KUMA removes the role from the list of roles, and permissions of the role are stripped from the users that had this role assigned. When a role is deleted while a session is active for users that have no other roles but this one, the sessions of such users are terminated. Assign other roles to such users or delete them.

Configuring the security policy for logging in to the web interface

To reduce the risk of unauthorized access to the KUMA web interface, you can configure rules for users logging in to the KUMA web interface. For example, configuring a security policy helps prevent an attacker from brute-forcing a password to log in to the KUMA web interface.

The security policy applies only to user accounts created in KUMA and does not affect domain accounts imported into KUMA when integrating with domain controllers. At the same time, the security policy affects the admin user account, which is used for the first login to the KUMA web interface, in a limited way. The only security policy setting that applies to this user account is the forced re-authentication requirement after a period of inactivity.

The security policy applies to all tenants and cannot be configured on a per-tenant basis.

When upgrading from a previous version of KUMA, the password requirements apply only to new passwords. Existing passwords are affected only by the **Password lifetime (days)**, other requirements do not apply to such passwords. For users whose passwords have not been changed for more than 90 days, KUMA suggests a password change after the upgrade.

Managing security policy settings requires the General administrator role.

To configure the security policy for logging in to the KUMA web interface

1. Open the **Settings** → **Access** → **Security policy settings** section of the KUMA web interface.
2. Specify the following security settings:

   - **Maximum number of failed authentications**. If this value is exceeded, the user account is locked out for a duration specified in the **Blockage period after failed authentication attempts (mins)** setting. Enter a number from 0 to 10. The default setting is 0, meaning that this setting is disabled.
   - **Maximum number of authentication attempts per second per Core**. This setting helps prevent brute-forcing a password to log in to the KUMA web interface. Enter a number from 1 to 8. The default value is 3. If KUMA detects that the specified value is exceeded, further authentication attempts are blocked.

     Note that the value is specified per KUMA Core. If your KUMA instance has more than one KUMA Core deployed, the actual number of allowed authentication attempts per second is calculated as the specified value multiplied by the number of KUMA Cores.
   - **Blockage period after failed authentication attempts (mins)**. The duration for which the user account is locked out when the value of the **Maximum number of failed authentications** setting is exceeded. Enter a number from 0 to 1440. The default value is 15. If 0, the account status is changed to *Inactive*. A user with this status cannot log in to the KUMA web interface. The General administrator can change the user account status later. If the General administrator changes the status of a user account to *Active*, the user account is unlocked and the counter of failed login attempts is reset to 0.

- **Prevent users from signing in after the password expiration**. This setting allows you to configure the behavior of the solution after the expiration of the **Password lifetime (days)**.

  If this setting is enabled, the user is locked out of the KUMA web interface. The user account status changes to *Inactive*. A user with this status cannot log in to the KUMA web interface. The General administrator can change the user account status later. The user is locked out if the user has the **User must change password at next login** setting enabled.

  If this setting is disabled, the user with an expired password can log in to the KUMA web interface. KUMA then automatically prompts the user to change the password.

- **Re-authentication is required after the user is inactive for (mins)**. If no mouse movement or key presses are registered for the specified time during a user's KUMA web interface session, KUMA requires the user to authenticate again. Enter a number from 1 to 1440. The default value is 15.

3. Specify the following user password settings:

   - **Password lifetime (days)**. After the password lifetime expires, KUMA locks out or does not lock out the user account, depending on the **Prevent users from signing in after the password expiration** setting. Enter a number from 1 to 365. The default value is 90.

   - **Minimum password length (symbols)**. Specify a number from 16 to 128. The default value is 16.

   - **Time to notify the user before the password expiration (days)**. When the specified number of days remains before the password expires, the user is prompted in the KUMA web interface to change the password. Enter a number from 1 to 14. The default value is 14.

   - **Character categories a password must include**. You can use this setting to configure requirements that apply to manually entered and automatically generated passwords:

     - The password must contain lowercase and uppercase letters of the Latin alphabet, numerals, and at least one of the special characters that you specified.

     - **Ignore repeated characters in the password**. If this check box is selected, two or more identical characters in a row are allowed in passwords.

4. Click **Save**.

Security policy is configured.

Creating a user account

To create a user account:

1. In the KUMA web interface, open **Settings** → **Access** → **Users**.

2. Click the **Create new** button and set the parameters as described below.

   - Enabling the **Active** toggle switch makes the user account active. The toggle switch is turned on by default.

     Only users with an *Active* account can log in to the KUMA web interface. The status of a user account can also change to *Inactive* automatically. For example, this might happen if the maximum allowed number of attempts to log in to the solution web interface is exceeded while the **Blockage period after failed authentication attempts** is set to 0.

   - In the **Name** field, enter the user name. The length of the string must be 1 to 128 Unicode characters.

- In the **Login** field, enter a unique user name for the user account. Must contain from 3 to 64 characters (only a–z, A–Z, 0–9, . \ - _).
- In the **Email** field, enter the unique email address of the user. Must be a valid email address.
- 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** to display timestamps in 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 **Notify about alerts in web interface** check box to display alert notifications in the KUMA web interface. Clear this check box if you want to disable notification display. This check box is selected by default.
- Select the **Display non-printable characters** check box to display non-printing characters such as spaces, tab characters, and line breaks in the KUMA web interface. 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 the **Description** field), 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 **New password** field, enter a new password. You can enter a password or generate one by clicking the **Generate password** button. The password must satisfy the following requirements:
  - The length must satisfy the security policy.
  - At least one lowercase character.
  - At least one uppercase character.
  - At lease one numeral.
  - At least one special character from the list of special characters in the security policy.
  - No more than two identical characters in a row.
  - The password does not contain the login.

The user must change the password when logging in to the KUMA web interface for the first time and subsequently change the password at least once every three months. The General administrator can manage both of these settings in the security policy.

- In the **Confirm password** field, enter the password again for confirmation.
- If the **User must change password at next login** check box is selected, the next time the user attempts to log in to the KUMA web interface, the user must specify the old password and set a new password. After that, the user is allowed to log in to the KUMA web interface with the new password. By default, this check box is selected.

  This setting is available only for *Active* accounts.

  You can use this setting, for example, when creating a user. You can assign a temporary password to the user and select this check box. After the user logs in with a temporary password, KUMA prompts the user to change the password.

  You can also make users reset their passwords in the **Settings** → **Access** → **Users** section by selecting users from the list and clicking the **Reset password at next login** button.
- Under **Access**, click the **Add role groups** buttons to assign roles to the user account in specific tenants. A user can be assigned different roles in different tenants; multiple roles can be assigned within the same tenant.
- 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.
- If the **Unlimited password lifetime** check box is selected, the **Password lifetime** setting does not apply to the user account.

  We recommend enabling this setting for a service account, restricting the rights of such an account, and setting a password of maximum complexity. Typically, a user account with an unlimited password lifetime is used for monitoring. When migrating users, you can manually enable this setting for those service accounts that were previously used to display the dashboard.

3. If necessary, use the **Generate token** button to generate an API token.

   Clicking this button displays the token creation window.
4. If necessary, configure the operations available to the user via the REST API by using the **API access rights** button.
5. Click **Save**.

The user account will be created and displayed in the **Users** table.

Editing a user account

To edit a user account:

1. In the KUMA web interface, open **Settings** → **Access** → **Users**.
2. Select the relevant user and in the user details area on the right, edit the settings as necessary.

If you want to change your own user account, you can also click the name of the user account in the lower left corner of the window to open a menu and click the **Profile** button in the menu.

- Selecting the **Active** check box makes the user account active. The toggle switch is turned on by default.

  A user cannot deactivate their own user account.

  Only users with an *Active* account can log in to the KUMA web interface. The status of a user account can also change to *Inactive* automatically. For example, this might happen if the maximum allowed number of attempts to log in to the solution web interface is exceeded while the **Blockage period after failed authentication attempts** is set to 0.

- In the **Name** field, edit the user name. The length of the string must be 1 to 128 Unicode characters.

- In the **Login** field, enter a unique user name for the user account. Must contain from 3 to 64 characters (only a–z, A–Z, 0–9, . \ - _).

- In the **Email** field, edit the unique email address of the user. The email address must be valid and unique.

- 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** to display timestamps in 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 **Notify about alerts in web interface** check box to display alert notifications in the KUMA web interface. Clear this check box if you want to disable notification display. This check box is selected by default.

- Select the **Display non-printable characters** check box to display non-printing characters such as spaces, tab characters, and line breaks in the KUMA web interface. If the **Display non-printable characters** check box is selected, you can press **`Ctrl/Command+*`** to enable and disable the display of non-printing characters.

  Spaces and tab characters are displayed in all input fields (except the **Description** field), 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.

- Under **Access**, use the **Add role groups** buttons to assign roles to the user in specific tenants. A user can be assigned different roles in different tenants; multiple roles can be assigned within the same tenant. For a domain user, the ability to change the main role (General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst) is blocked in the user card, while additional roles can be added or removed (Access to CII, Interaction with NCIRCC, Read shared resources), including management of additional role assignment to tenants.

- 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.

3. If you need to change the password, click the **Change password** button and fill in the fields described below in the opened window. When finished, click **OK**.

   - In the **Old password** field, enter the current password of the user account. The field is available if you change your account password.

   - In the **New password** field, enter a new password. You can enter a password or generate one by clicking the **Generate password** button. The password must satisfy the following requirements:

     - The length must satisfy the security policy.
     - At least one lowercase character.
     - At least one uppercase character.
     - At lease one numeral.
     - At least one special character from the list of special characters in the security policy.
     - No more than two identical characters in a row.
     - The password is different from the last three passwords of this account.
     - The password does not contain the login.

   - In the **Confirm password** field, enter the password again for confirmation.

   - If the **User must change password at next login** check box is selected, the next time the user attempts to log in to the KUMA web interface, the user must specify the old password and set a new password. After that, the user is allowed to log in to the KUMA web interface with the new password. This check box is selected by default.

     This setting is available only for *Active* accounts.

4. If necessary, use the **Generate token** button to generate an API token. Clicking this button displays the token creation window.

5. If necessary, configure the operations available to the user via the REST API by using the **API access rights** button.

6. If the **Unlimited password lifetime** check box is selected, the **Password lifetime** setting does not apply to the user account.

   We recommend enabling this setting for a service account, restricting the rights of such an account, and setting a password of maximum complexity. Typically, a user account with an unlimited password lifetime is used for monitoring. When migrating users, you can manually enable this setting for those service accounts that were previously used to display the dashboard.

7. Click **Save**.

The user account will be changed.

Special considerations involved in changing the set of spaces assigned to a user

A user is assigned one or more space sets; information about these is stored in the user account settings. To view spaces in a space set, hover over the space set in the user profile. If necessary, you can add or remove spaces from the space set in the **Settings** → **Access** → **Access to spaces** section.

If a space is deleted from the storage, the space is still displayed as part of the space set with the `deleted+UUID` designation. In this case, only the first 8 characters of the space's universal unique ID (UUID) are displayed. A deleted space is longer displayed as part of the space set after you save changes to that space set.

> A space is completely deleted from the space set if that space has no events. If there are some events in the space being deleted, KUMA displays the space as `deleted+UUID` even after saving changes to the space set.

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 in the KUMA web interface based on a resource set 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.
- Event routers receive streams of events from collectors and correlators and then distribute the events to destinations in accordance with the configured filters.

In the KUMA web interface, services are displayed in the **Resources** → **Active services** section in table format. The table of services can be updated using the **Refresh** button and sorted by columns by clicking on the active headers. You can also configure the columns displayed in the table. To do so, click the gear button in the upper right corner to display a drop-down list. In that drop-down list, select check boxes next to the names of the columns that you want to display in the table. You can leave any single column in the list to be displayed.

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 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.
- **Type**—type of service: **agent**, **collector**, **correlator**, **storage**, **event router**.
- **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**—Uptime of the service.
- **Created**—the date and time when the service was created.
- **UUID**—Unique ID of the service.

  By default, this column is not displayed in the table. You can add it to the table by clicking the gear icon in the upper right part of the table of services and selecting the check box next to the name of the **UUID** column in the drop-down list.

You can sort data in the table in ascending and descending order, as well as by the **Status** parameter and by the service type in the **Type** column. To sort active services, right-click the context menu and select one or more statuses and a type.

You can use 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**

  The **Update settings** button is not available if the KUMA Core service is among the services selected for group actions or if any of the selected services has the grey status. To make the **Update settings** button available for group actions, clear the check box from the KUMA Core service and services with the grey status.
- 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 web interface.

Renaming a service

If necessary, you can change the name and settings of created services.

To rename a service:

1. In the web interface, open the **Resources** → **Active services** section.
2. Click the link with the name of the service in the **Service** column.
3. In the window that opens, specify a new service name and, if necessary, change the settings. The name must be unique within a tenant and contain 1 to 128 Unicode characters.

   For the KUMA Core and Metrics services, only the name can be changed.
4. Click the **Save** button.

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 web interface and open **Resources** → **Active services**.
2. In the context menu of the name of the service whose ID you want to get, select **Copy ID**.

The identifier of the service will be copied to the clipboard. For instance, this ID can be used to install 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 while managing KUMA.

Commands for stopping, starting, and checking the status of a service

Service	Stop service	Start service	Check the statu
Core	`sudo systemctl stop kuma-core-<service ID>.service`	`sudo systemctl start kuma-core-<service ID>.service`	`sudo systemctl statu <service ID>.servic`
Services with an ID: • collector • correlator • storage • metrics	`sudo systemctl stop kuma-<collector/correlator/storage/metrics>-<service ID>.service`	`sudo systemctl start kuma-<collector/correlator/storage/metrics>-<service ID>.service`	`sudo systemctl statu <collector/correlato <service ID>.servic`
Windows agents	To stop an agent service: 1. Copy the agent ID in the KUMA web interface. 2. Connect to the host on which you want to start the KUMA agent service. 3. Run PowerShell as an account with administrative rights. 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 web interface. 2. Connect to the host on which you want to start the KUMA agent service. 3. Run PowerShell as an account with administrative rights. 4. Run the following command in PowerShell: `Start-Service -Name "WindowsAgent-<agent ID>"`	To view the status of an a 1. In Windows, go to the S and in the list of services, relevant KUMA agent. 2. This opens a window; in status of the agent in the

Restarting the service

To restart the service:

1. In the KUMA web interface, open **Resources** → **Active services**.
2. Select the check box next to the service and choose one of the following options:
   - **Update configuration**—perform a hot update of a running service configuration. For example, this way you can change field mapping settings or destination point settings.
   - **Restart**—stop a service and start it again. This option is used to modify such parameters, as a 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 computer. 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 have selected **Reset certificate**, KUMA generates an error stating that the agent is not running. In the `%PROGRAMDATA%\Kaspersky Lab\KUMA\agent\<agent ID>\certificates` agent installation folder, manually delete `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 `/opt/kaspersky/agent/<Agent ID>/certificates` agent installation folder, manually delete `internal.cert` and `internal.key` files.

  3. Since using the **Reset certificate** option stops the agent, you need to start the agent manually to continue its operation. 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 web interface:

1. Log in to the KUMA web interface and open **Resources** → **Active services**.

2. In the context menu of the service name, select **Delete**.

3. In the confirmation window, 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/eventRouter/metrics> --core https://<FQDN of the KUMA Core server>:<port that KUMA Core uses for internal communications (7210 by default)> --id <service ID> --
```

```
uninstall
```

```
For example:
```

```
sudo /opt/kaspersky/kuma/kuma collector --core https://kuma.example.com:7210 --id 29fd6760-a870-42d2-8ebe-eb911ac90305 --uninstall
```

The service is deleted from the server.

Viewing the storage partition table

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 web interface and open **Resources** → **Active services**.
2. Select the check box next to the relevant storage.
3. In the toolbar or the menu (⋮), click **Go to partitions**.

   The **Partitions** table opens.

The resource table contains the following columns:

- **Tenant** is the name of the tenant that owns the stored data.
- **Created** is the partition creation date.
- **Space** is the name of the space.
- **Size** is the size of the space.
- **Events** is the number of stored events.
- **Transfer to cold storage** is the date when data will be migrated from the ClickHouse clusters to cold storage disks.
- **Expires** is the date when the partition expires. After this date, the partition and the events it contains are no longer available.

You can filter partitions in the table by tenant and space. To do this, click the header of the relevant column and select filters from the drop-down list. The set of filters available in the drop-down list depends on the selected column.

You can delete partitions.

To delete a partition:

1. Open the **Partitions** table (see above).
2. Select the check box next to the storage you want to delete, and click **Delete**.

   A confirmation window opens.
3. Click **OK**.

The partition is deleted. Audit event partitions cannot be deleted.

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 KUMA web interface, go to the **Resources** → **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 **ClickHouse active queries** table. If you have selected 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 KUMA 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 by clicking 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 the query that you want to cancel and click **Cancel query**. You can select one or more queries.

   A confirmation window opens.
3. If necessary, in the **Specify a reason** field, specify the reason why you are canceling the request.
4. Click **OK**.

The query is canceled and removed from the table. An audit event is automatically generated.

Searching for related events

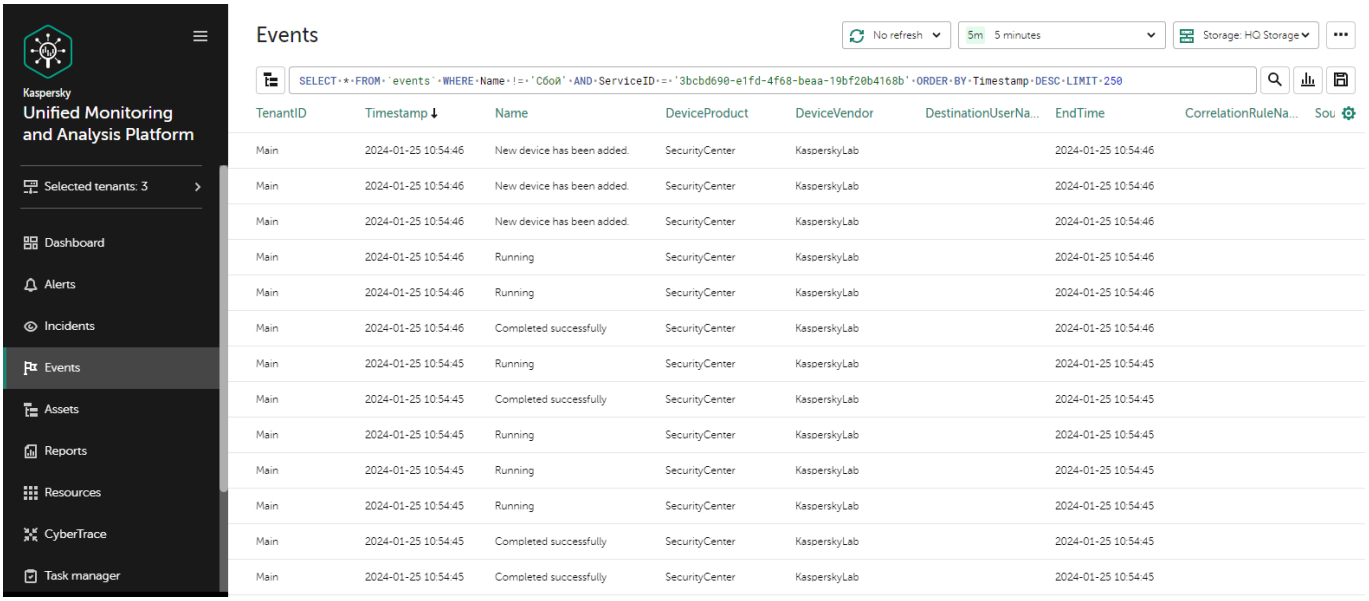
You can search for events processed by the Correlator or the Collector services.

To search for events related to the Correlator or the Collector service:

1. Log in to the KUMA web interface and open **Resources** → **Active services**.
2. Select the check box next to the required correlator or collector and click **Go to Events**.

   This opens a new browser tab 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.



Event search results

When searching for events, you may encounter the following shard unavailability error:

```
Code: 279. DB::NetException: All connection tries failed. Log: \\n\\nTimeout exceeded
while connecting to socket (host.example.com:port, connection timeout 1000 ms)\\nTimeout
exceeded while connecting to socket (host.example.com:port, connection timeout 1000
ms)\\nTimeout exceeded while connecting to socket (host.example.com:port, connection
timeout 1000 ms)\\n\\n: While executing Remote. (ALL_CONNECTION_TRIES_FAILED) (version
23.8.8.207)\\n\"}",
```

In this case, you need to override the ClickHouse configuration in storage settings.

To override the ClickHouse configuration:

1. In the KUMA web interface, select the **Resources** section.
2. In the **Services configuration** → **Storages** section, click the storage resource that you want to edit.

   This opens the **Edit storage** window.
3. In the **ClickHouse configuration override** field, specify the following settings:

   - To skip unavailable shards when searching, insert the following lines:

     ```
     <profiles>
             <default>
                     <skip_unavailable_shards>1</skip_unavailable_shards>
             </default>
     </profiles>
     ```

   - To set memory usage limits, specify `max_server_memory_usage` or `max_memory_usage_for_all_queries` parameters as recommended.
4. To apply the ClickHouse configuration, click **Save**.
5. Restart the storage services that depend on this resource.

   This resolves the shard unavailability error, and you can proceed to search for events processed by a particular correlator or collector.

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 resource set must be owned by the same tenant that owns the created resource set. 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 web interface. 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 web interface, 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.

We recommend using the ext4 file system.

A storage is created in several steps:

1. **Creating a set of resources for a storage in the KUMA web interface**
2. **Creating a storage service in the KUMA web interface**
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.

The storage service can be started if IPv6 and IPv4 formats are allowed. If you only use IPv4, specify the following in the `override configuration` field of the `storage` resource:

```
<interserver_listen_host>0.0.0.0</interserver_listen_host>
```

After changing the settings of a storage, its 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's 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, we no longer recommend using HDFS as cold storage. Support for the HDFS file system will be discontinued in a future release. Users that use HDFS are advised to plan a migration to S3.

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.

The storage service may fail and cannot be started again in the following cases:

- If a cold storage disk is not configured and the server runs out of disk space in hot storage.
- If a cold storage disk is configured and space on it runs out.
- If the cluster cannot gain access to the configured cold storage disk.

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.
The storage service must also be restarted if the path to the S3 storage was 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 remove 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 disks of the ClickHouse cluster are 97% full, the largest partitions are either automatically transferred to cold storage disks, or deleted if there are no cold storage disks.

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:

1. In the web interface, go to the **Resources** → **Storages** section and select the storage.
2. In the window that opens, in the **Disks for cold storage** settings section, in the required disk settings section, click the X icon to remove the disk.
3. In the confirmation window, click **OK**.

Data from removed disk is automatically migrated to other cold storage disks or, if there are no such disks, to the ClickHouse cluster. While data is being migrated, the status icon of the storage turns yellow and an hourglass icon is displayed. 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 delete the data partitions manually using the following command:

```
ROP PARTITION
```

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 and S3 disks) and then remove 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.

You can also use the created partition archives as backup copies of these partitions.

Detaching partitions

To disconnect partitions:

1. Select the shard on all replicas of which you want to disable the partition.
2. Get the partition ID using the following command:

   ```
   sudo /opt/kaspersky/kuma/storage/<storage ID>/deps/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 received ID:

   ```
   sudo /opt/kaspersky/kuma/storage/<storage ID>/deps/clickhouse/bin/client.sh -d kuma --multiline --query "ALTER TABLE events_local_v2 DETACH PARTITION ID '<partition ID>'"
   ```

The partition is disabled on all replicas of the shard. Now you can move data 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. Go to the `detached` directory with a disconnected partition and do the archiving:

   ```
   sudo cd <path to the detached directory containing the detached partition>
   sudo zip -9 -r detached.zip *
   ```

   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 connect archive partitions to the solution:

1. In the web interface, open the **Resources** → **Storage** section and in the settings of the used storage, increase the value of the **Event retention time** setting.

   Data is deleted according to the date specified in the Timestamp field (when the event was received) and according to the value of the **Event retention time** setting that you specified for the storage.

   Before restoring archived data, make sure that the **Event storage period** value is greater than 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 into the `detached` directory of your storage, and extract the archive:

   ```
   sudo unzip detached.zip -d <path to the detached directory>
   ```

   Example

   ```
   sudo unzip detached.zip -d /opt/kaspersky/kuma/clickhouse/data/store/d5b/d5bdd8d8-e1eb-4968-95bd-d8d8e1eb3968/detached/
   ```

3. Connect the partition using the command:

   ```
   sudo /opt/kaspersky/kuma/storage/<storage ID>/deps/clickhouse/bin/client.sh -d kuma --multiline --query "ALTER TABLE events_local_v2 ATTACH PARTITION ID '<partition ID>'"
   ```

4. Repeat the steps of extracting the archive and attaching the partition on each replica of the shard.

The archived partition is attached and its events are again available for search.

Creating a set of resources 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 remove 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 a retention time for events:

   - **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.

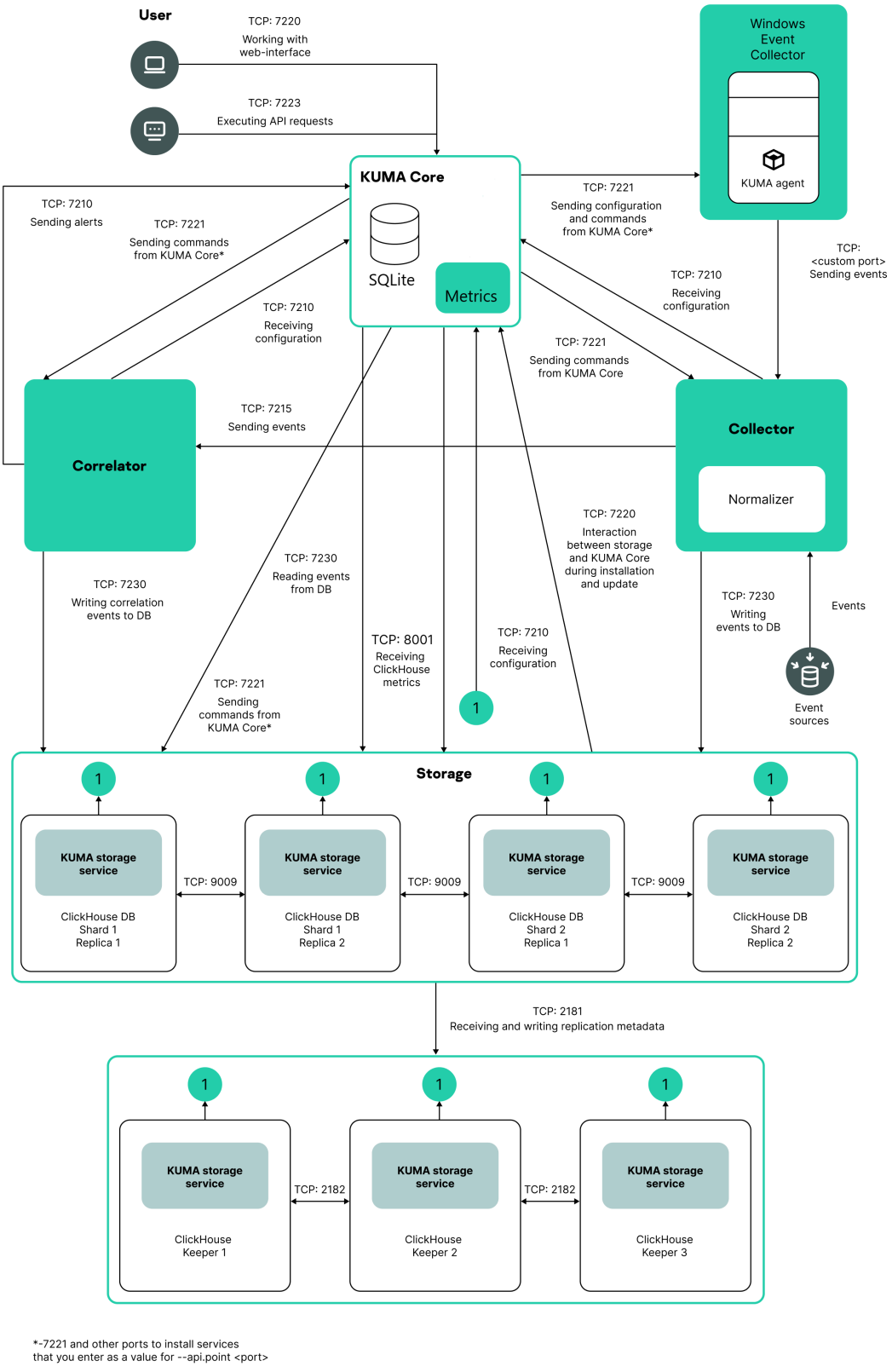
12. If necessary, in the **ClickHouse cluster nodes** section, add ClickHouse cluster nodes to the storage.

    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 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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 storage disks. Support for the HDFS file system will be discontinued in one of future releases of KUMA.

- In the **Name** field, specify the disk name.

  Rules for naming an S3 storage

  When assigning a name to an S3 storage, the following requirements must be met:

  - The name must be unique within a single storage.
  - The name may include lowercase Latin letters and numerals. You can also use the underscore character (_) in the name.

    Non-Latin letters are not allowed.
  - The name must contain 1 to 128 Unicode characters.

  You cannot change the name of an S3 storage disk without changing the path to the disk.

- If you select the **Local** domain name type for the disk, specify the absolute directory path of the mounted local disk in the **Path** field. The path must begin and end with a "/" character.

  The directory of the mounted local disk must be owned by the KUMA user.

- 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 drive, 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 the address of the S3 storage

    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 storage 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**.
- In the **Certificate chain** field, specify the path to the file with the complete certificate trust chain for connecting to the S3 storage.

  You must fill in this field if you are connecting to the S3 storage via HTTPS. If you are using HTTP, you do not need to specify a certificate. The file must contain the full BASE64-encoded certificate trust chain, including the root certificate, intermediate (if any), and S3 storage certificate.

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. KUMAIf 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 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 web interface

When a resource set is created for a storage, you can proceed to create a storage service in KUMA.

To create a storage service in the KUMA web interface:

1. In the KUMA web interface, under **Resources** → **Active services**, click **Add service**.
2. In the opened **Choose a service** window, select the resource set that you just created for the storage and click **Create service**.

   The storage service is created in the KUMA web interface 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

For the storage service to work correctly, the host name of the server must match one of its IP addresses. Review the /etc/hosts file to make sure it matches. You can view the list of local IP addresses using the `ip a` command.

To create a storage:

1. Log in to the server where you want to install the service.
2. 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 web interface> --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`.

3. Repeat steps 1–2 for each storage node.

> Only one storage service can be installed on a host.

The storage is installed.

Recommendations for using system memory

To regulate memory consumption when creating a storage on systems with limited memory, we recommend determining the current memory usage and increasing the `max_server_memory_usage` value. This setting determines the maximum amount of memory that the server can use. This value is specified in bytes.

By default, `max_server_memory_usage` is set to 0, which allows the server to use all available memory.

The `max_server_memory_usage` value is usually further limited by the `max_server_memory_usage_to_ram_ratio` value (the maximum amount of memory that a server can use as a ratio to all available memory). For example, the default value of 0.9 means that the server can consume 90% of the available memory.

When determining the current memory usage, keep in mind the following limitations:

- The optimal amount of system memory available to the storage is 75% of the total memory.

  > We do not recommend allowing all 100% of the available memory to be used up. This can exhaust RAM when processing incoming requests.

- If other applications are installed on the server, you need to leave enough memory to them.
- If the hardware of the host server changes while the storage is in use, you need to be mindful of the configured value and adjust it accordingly.

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 KUMA web interface, go to the **Resources and services** → **Active services** section and open an active storage to manage its settings.

2. In the **Edit storage** window that opens, on the **Basic settings** tab, in the **Backup and recovery settings** 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. Make sure that the 'kuma' user has access rights to the specified directory.

   If no path is specified, performing the backup is impossible.

   b. Enable the **Add automatic 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.

   If multiple shards are used in the cluster, the **Replicas** list contains only those replicas whose IDs are specified in all shards.

3. 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.

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 KUMA web interface, in the **Resources** → **Active services** section, open the current storage to manage its settings.

2. This opens the **Edit storage** window; in that window, on the **General settings** tab, in the **Backup and restore 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.

   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/`

3. In the KUMA web interface, in the **Resources** → **Active services** section, select the check box next to the storage service that has the green status.

4. In the toolbar or in the menu (⋮), select **Backup** and click **Create a backup copy**.

   The **Create a backup copy** 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 **Date range** 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.

Restoring partitions from backup

Only users with the General Administrator role can restore partitions from backup.

To restore a partition from backup:

1. Move the zip file of the backup copy from the **backup** directory to the **restore** directory.

2. Grant the 'kuma' user permissions to the zip file of the backup copy in the restore directory using the following command:

   ```
   sudo chown kuma *.zip
   ```

3. In the KUMA web interface, in the **Resources** → **Active services** section, select the check box next to the storage service that has 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.

Importing partitions from an external KUMA installation

Users with the General Administrator role can import partitions from an external KUMA installation.

To import partitions from an external KUMA installation:

1. Place the zip file from which you want to import partitions into the **restore** directory. Partitions will be imported from this directory.

   Grant the 'kuma' user permissions to the zip file of the backup copy in the restore directory using the following command:

   ```
   sudo chown kuma *.zip
   ```

2. In the KUMA web interface, in the **Resources** → **Active services** section, select the check box next to the storage service that has the green status.

3. In the toolbar or the menu (⋮), select **Backup** → **Import**.

4. 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

5. Click **OK**.
6. 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.

Creating a correlator

A correlator consists of two parts: one part is created inside the KUMA web interface, and the other part is installed on the network infrastructure server intended for processing events.

Actions in the KUMA web interface

A correlator is created in the KUMA web interface by using the Installation Wizard, which combines the necessary resources into a resource set for the correlator. Upon completion of the Wizard, the service is automatically created based on this resource set.

To create a correlator in the KUMA web interface:

Start the Correlator Installation Wizard:

- In the KUMA web interface, under **Resources**, click **Create correlator**.
- In the KUMA web interface, under **Resources** → **Correlators**, click **Add correlator**.

As a result of completing the steps of the Wizard, a correlator service is created in the KUMA web interface.

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 identifier automatically assigned to the service in the KUMA web interface, 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 web interface, under **Resources**, click **Add correlator**.
- In the KUMA web interface, 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 using 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 web interface 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 specify the general settings of the correlator:

1. On the **Basic settings** tab, fill in the following fields:

   - 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 the Shared tenant can be added to the service.
   - In the **Workers** field, if necessary, specify the number of workers that the service can run simultaneously to process events in parallel. Specify a positive integer number from 0 to 999. If 0 is specified, the number of logical CPU cores is used.
   - You can optionally add up to 256 Unicode characters describing the service in the **Description** field.

2. On the **Advanced settings** tab, fill in the following fields:

   - If necessary, use the **Debug** toggle switch to enable 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. The specifics of filling in this field must be provided by Technical Support.
   - In the **Number of incoming requests in a queue in a queue** field, specify the maximum number of database insert queries that can be queued for processing by the correlator.

General settings of the correlator are specified. 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.

  When entering functions, you can use autocomplete as a list of hints with possible function names, their brief description and usage examples. You can select a function from the list and insert it together with its list of arguments into the input field.

  To display the list of all hints in the field, press **`Ctrl+Space`**. Press **`Enter`** to select a function from the list. Press **`Tab`** to go to the next argument in the list of arguments of the selected 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 resource set 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.
- **Asset category change**—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 the Shared tenant.

You can use the **Search** field to search for a correlation rule. Added correlation rules can be removed from the resource set 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 resource set.

Use 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. Using 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 resource set 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 resource set for the correlator and will be displayed in the rules table.

To create a new correlation rule in a resource set 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 resource set 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 resource set:

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 resource set for the correlator.

To create a new enrichment rule in a resource set:

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 enrichment fields** setting 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 extended schema array 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.
  - Under **Conversion**, 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 for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, 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.

- Converting the IP address to the IPv4 format:

  - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.

  - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

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 the result of 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** is the number of workers that the service can run simultaneously to process events in parallel. Specify a positive integer number from 0 to 999. If 0 is specified, the number of logical CPU cores is used.

- **Max tasks**—maximum number of simultaneously fulfilled requests. By default, this value is equal to the number of vCPUs of the KUMA Core server.

- **Cache TTL**—the lifetime (in seconds) of the values stored in the cache. The default value is `60` (seconds).

- **Cache disabled**—you can use this toggle switch to enable or disable caching. Caching is enabled by default.

- **Cache update disabled**—if cache update is enabled, then after the cache TTL expires, the collector sends repeat queries to the DNS server to update the information stored in the cache. This setting is available only when cache is enabled. By default, cache update is enabled.

- The **Recursive queries** setting is available starting with KUMA 3.4.1. You can use this toggle switch to make a KUMA collector send recursive queries to authoritative DNS servers for the purposes of enrichment. The default value is **Disabled**.

- **cybertrace**

  For systems with a high load on the collector or correlator, we recommend using the cybertrace-http enrichment type. This type returns a structured response in JSON format and correctly distinguishes between strings and arrays of custom attributes, preventing parsing errors.

  When enriched through cybertrace, the collector gets an unstructured response. If the value of a custom attribute in CyberTrace contains square brackets, for example, **[str1**, **str2]**, KUMA treats it as an array, not a string.

  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.

  - **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 is a new streaming event enrichment type in CyberTrace that allows you to send a large number of events with a single request to the CyberTrace API. Recommended for systems with a lot of events. The performance of cybertrace-http is superior to that of the cybertrace type.

Limitations:

- The cybertrace-http enrichment type cannot be used for retroscan in KUMA.
- If the cybertrace-http enrichment type is being used, detections are not saved in CyberTrace history in the Detections window.

Available settings:

- **Name** (required) is a unique name for the enrichment rule. Up to 128 Unicode characters.
- **Tenant** (required) is the name of the tenant in which you plan to work with CyberTrace.
- **URL** (required)—in this field, you can specify the URL of a CyberTrace server to which you want to send requests and the port that CyberTrace API is using. The default port is 443.
- **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 1,000,000. 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.
- **Major version of CyberTrace** selects 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**.
- The **Debug** toggle switch enables resource logging. The toggle switch is turned off by default.
- **Tags** for resource search.

- **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.

- **check DLL Hijacking**

  DLL Hijacking is an attack technique that involves delivering vulnerable legitimate software along with a malicious dynamic link library (DLL) to the target system. 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 status, KUMA sends a request to KSN with information from event fields. In response to the request, KSN sends the analysis result.

  You can configure DLL Hijacking attack detection in the collector or correlator.

  > To use enrichment such as **check DLL Hijacking**, your license must include the AI module. The General administrator must also accept the additional KSN Statement in the **Kaspersky Security Network** section (**Settings** → **Integrations** → **AI services** → **DLL Hijacking detection**).

Available settings:

- **Workers** is the number of workers that the service can run simultaneously to process events in parallel. Specify a positive integer number from 0 to 999. If 0 is specified, the number of logical CPU cores is used.

- **Maximum cache size** is the maximum cache size for KSN responses, in megabytes. The default value is `128 MB`.

  Each KSN response has its own lifetime, during which it is retained in the cache. If KUMA sends a second request for a DLL file for which KSN has already issued a response, and the lifetime of that response has not yet expired, the request is not actually sent to KSN, but the response is taken from the cache.

- If the **Maximum number of events in enrichment queue** is exceeded, new events are not processed. The default value is `1,000,000`. `0` means that event processing stops if the queue contains any number of events.

- **Perform a repeat request**. KSN may return the **Unclassified** status. This status means that KSN has accepted the request and is currently processing it, but there is no status yet. Processing may take a long time, so the request must be re-sent. If this setting is enabled, specify additional settings:

  - If the **Create duplicates** setting is enabled, KUMA creates a duplicate event and uses it for repeated requests to KSN. In this case, the main event is routed in accordance with the workflow. A duplicate event is written to the database when one of the following conditions is satisfied:

    - A status other than **Unclassified** is received from KSN.
    - The disk buffer configured using the **Buffer size** setting is full.
    - The **Request count** is exceeded.

    If this setting is disabled, KUMA continues to send requests using the main event. The event is written to the database when one of the following conditions is satisfied:

    - A status other than **Unclassified** is received from KSN.
    - The disk buffer configured using the **Buffer size** setting is full.
    - The **Request count** is exceeded.

  - The **Request count** is the maximum number of requests to KSN. If, after the specified number of requests, a status other than **Unclassified** is not received, the event is written to the database with the **Unclassified** status. The default value is `10`.

  - **Poll interval** is the interval between requests, in minutes. The default value is `1 minute`. The minimum value is `1 minute`.

  - **Buffer size** is the size of the disk buffer for events awaiting a repeat request, in megabytes. The default value is `10240 MB`. The minimum value is `10 MB`.

- **Proxy** specifies proxy server settings, if one is needed to connect to the KSN server. You can select one of the available proxy servers or create a new proxy server. By default, a proxy server is not used.

- **Mapping** is a table of parameters that must be sent in a KSN request to get a status, and a list of the corresponding KUMA event fields. List of parameters in the request:

  - **dll hash (md5)**. Recommended field for mapping: OldFileHash.

  - **dll path**. Full path including the file name; recommended field for mapping: OldFilePath. The path may contain personal data, so KUMA sends this parameter to KSN in an anonymized form. For example:

    Path before anonymization: `C:\Windows\Device\HarddiskVolume1\Users\Alex\malware.dll`

    The same path in anonymized form: `device\harddiskvolume1\users\username`

  - **process hash (md5)**. Recommended field for mapping: FileHash.

  - **process path**. Full path including the file name. The path may contain personal data, so KUMA sends this parameter to KSN in an anonymized form. For example:

    Path before anonymization: `C:\Windows\Device\HarddiskVolume1\Users\Alex\malware.dll`

    The same path in anonymized form: `device\harddiskvolume1\users\username`

  - **dll name**. Recommended field for mapping: OldFileName.

  - **process name**. Recommended field for mapping: FileName.

  - **process signature**. Optional; recommended field for mapping: FileID. In this field, you need to specify the KUMA field that passes the Organization from the Subject of the signature. If these values are not available, leave the signature empty. To add this parameter, click the **Add** button, then in the **DLL Hijacking field** field, select **process signature** from the drop-down list.

  - **dll size**. Optional; recommended field for mapping: OldFileSize. To add this parameter, click the **Add** button, then in the **DLL Hijacking field** field, select **dll size** from the drop-down list.

  For each parameter, specify a corresponding KUMA event field in the table.

4. Use the **Debug** toggle switch 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

   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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

The new enrichment rule was added to the resource set 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 specify actions that 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 resource set:

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 resource set for the correlator.

To create a new response rule in a resource set:

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:

- **Response via KSC**—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

  - **Kaspersky Security Center task** (required)—name of the Kaspersky Security Center 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 Kaspersky Security Center tasks that can be started using KUMA:

    - Update
    - Virus scan

  - **Event field** (required)—defines the event field of the asset for which the Kaspersky Security Center task should be started. Possible values:

    - SourceAssetID
    - DestinationAssetID
    - DeviceAssetID

  To send requests to Kaspersky Security Center, you must ensure that Kaspersky Security Center 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 directory, the service will not start.

  - **Script arguments**—parameters or correlation 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 (").

    Correlation 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}}"`

- **Response via KEDR**—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 KUMA 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 application web interface. In the **Created by** column of the task table, **Scheduled task** is displayed for this 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 application 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.

- **Response via KICS/KATA**—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. Possible values:
    - Add account to group
    - Remove account from group
    - Reset account password
    - Block account

4. In the **Workers** field, specify the number of workers that the service can run simultaneously to process events in parallel. Specify a positive integer number from 0 to 999. If 0 is specified, the number of logical CPU cores is used.

   This field is optional.

5. 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

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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

The new response rule was added to the resource set 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 resource set 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 solution.

   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 using 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 resource set 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 solution.

   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.
   - Use the **Disabled** toggle button to specify whether events will be sent to this destination. By default, sending events is enabled.

- Select the **Type** of the destination:
  - 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 using 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. Default value: 67,108,864 (64 MB), displayed as 0 in the interface.
   - **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 solution 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** is the number of workers that the service can run simultaneously to process events in parallel. Specify a positive integer number from 0 to 999. If 0 is specified, the number of logical CPU cores is used.

- You can set health checks using the **Health check path** and **Health check timeout**. 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

  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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

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 resource set for the correlator is displayed under **Resources** → **Correlators**. It can be used to create new correlator services. When this resource set changes, all services that operate based on this resource set will start using the new parameters after the services restart. To do so, you can use the **Save and restart services** and **Save and update service configurations** buttons.

  A resource set 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 solution. 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 resource set 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 service must be installed on the server intended for processing events.

Installing the correlator on the server

A correlator consists of two parts: one part is created inside the KUMA web interface, 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. 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 web interface> --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 web interface, open **Resources** → **Active services**.
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 a correlator-ng correlator

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.

KUMA 4.2 introduces a beta version of correlator-ng for technical reference. Methods, algorithms, syntax of conditions, and other functions and characteristics may change before the release of the fully functional version of the correlator API.

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 correlator-ng

KUMA 4.2 introduces a beta version of correlator-ng for technical reference. Methods, algorithms, syntax of conditions, and other functions and characteristics may change before the release of the fully functional version of the correlator API.

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 begins to process events.

The correlator can be deployed:

- On a single server
- In a RAFT cluster for high availability and scalability

This section describes the structure of the configuration of the service and the purpose of the configuration files.

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.

Deploying correlator-ng in a Raft cluster

KUMA 4.2 introduces a beta version of correlator-ng for technical reference. Methods, algorithms, syntax of conditions, and other functions and characteristics may change before the release of the fully functional version of the correlator API.

You can deploy correlator-ng in a Raft cluster. The cluster synchronizes the state of the correlation buckets between nodes and event processing to continue events when individual nodes fail.

We recommend using an odd number of nodes to ensure high availability.

The minimum number of available nodes that a cluster needs to function can be calculated using the formula: *N / 2 + 1*, where N is the total number of cluster nodes. For example, if you have 3 nodes, the tolerable number of unavailable nodes is 1, with 5 nodes it is 2, and with 7, it is 3.

Each correlator node is registered in the KUMA Core as a separate service.

Command line options of the correlator

Correlator-ng is started on the command line and accepts arguments.

Command line options of the correlator

Setting	Description
`--id`	Unique ID of the correlator service
`--core`	FQDN of the KUMA Core service, for example, `https://kuma-core.example.com:7210`.
`--api.port`	API port of the correlator.
`--wd`	Working directory of the correlator.
`--advertise.fqdn`	FQDN of the node that will be passed to the KUMA Core service for network access to the correlator.
`--advertise.api.port`	API port that will be passed to the KUMA Core service.
`--san`	List of subject alternative names for the correlator.
`--http.client.connection.timeout`	HTTP client timeout when communicating with KUMA microservices, in seconds. The default setting is `10`.
`--logger.filter`	Logging level of the correlator. Possible values: `error`, `warn`, `info`, `debug`, `trace`. The default setting is `info`.
`--threads.async`	The number of threads for asynchronous operations (network operations and signal processing). The default setting is `1`.

Options of the Raft cluster

Options of the Raft cluster

Setting	Description
`--raft.tick-interval`	Maximum time interval for the node to react to changes in the cluster, in milliseconds. The default setting is `500`.
`--raft.heartbeat.tick`	Period of messages exchange between cluster nodes, in milliseconds. The default setting is `2`.
`--raft.election.tick`	Time interval after which the node initiates the election of the control node, in milliseconds. The default setting is `2`.
`--raft.proposition.tick`	Number of time intervals before the accumulated changes are sent to other nodes in the cluster. The default setting is `2`.
`--raft.proposition.max-size`	The maximum size of the batch of changes sent to other cluster nodes, in megabytes. The default setting is `4`.
`--raft.max-commited-size-per-tick`	Maximum amount of data that can be received and processed by the node in one time interval, in megabytes. The default setting is `64`.
`--raft.request.timeout`	Timeout for sending Raft requests, in seconds. The default setting is `10`.
`--raft.entries-before-snapshot`	The number of entries in the Raft log before a snapshot is taken. The default setting is `1000`.
`--raft.shards`	Number of Raft shards for distributing data between groups. When changing the setting, a full reset of the correlator state is required. The default setting is `4`.
`--raft.join`	FQDN of the cluster node to which the new correlator node connects.

Advanced settings

Advanced settings

Setting	Description
`--suppress-alerts`	Disables the sending of correlation events to the KUMA Core service.

Deploying the first node

Before creating a cluster, you must start the first node of correlator-ng.

To start the first node, you must specify the required options:

- `--id`, the unique ID of the correlator service
- `--core`, the FQDN of the KUMA Core service
- `--wd`, the working directory of the correlator
- `--api.port`, the API port of the correlator.

Example of starting on the command line:

```
/opt/kaspersky/kuma/kuma ng correlator --id <correlator_service_ID> --core <FQDN_of_KUMA_Core_service> --wd
<working_directory_of_correlator> --api.port <API_port_of_correlator>
```

After starting, the following directories are created in the working directory of the correlator:

- certificates
- config
- containers
- correlation

After receiving the configuration from the KUMA Core service, errors may occur related to the strong typing of selector expressions in the correlation rules. In this case, you need to fix the errors in the relevant YAML files in the config directory by following the instructions of the error messages.

After successfully starting the first node and fixing any configuration errors, you can proceed to deploying the remaining nodes of the cluster.

Connecting nodes to the Raft cluster

To add nodes to the Raft cluster:

1. Create the working directory of the correlator:

   ```
   mkdir /opt/correlator-wd
   ```

2. Copy the 'config' directory from the working directory of the first correlator node to the working directories of the other nodes.

3. Run correlator-ng with the `--raft.join` option to join the cluster:

```
/opt/kaspersky/kuma/kuma ng correlator --wd <working_directory_of_correlator> --api.port
<API_port_of_the_first_correlator_node> --id <KUMA_service_ID> --core <FQDN_of_KUMA_Core_service> --raft.join
<FQDN_of_first_correlator_node>:<API_port_of_first_correlator_node>
```

You need to run the launch command on each additional correlator node by changing the `--id` (unique ID of the KUMA service) and `--wd` (working directory of the correlator node) options.

The `--raft.join` option can point to any node that is already a member of the cluster. Specify the address of the first correlator node is not necessary.

Note that each correlator node has its own KUMA service ID. Therefore, to deploy the cluster, you need to create an appropriate number of correlator services in the KUMA web interface. All correlator services must be based on the same correlator resource.

Supported data types for correlator-ng

KUMA 4.2 introduces a beta version of correlator-ng for technical reference. Methods, algorithms, syntax of conditions, and other functions and characteristics may change before the release of the fully functional version of the correlator API.

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)
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

Data type	Description
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

KUMA 4.2 introduces a beta version of correlator-ng for technical reference. Methods, algorithms, syntax of conditions, and other functions and characteristics may change before the release of the fully functional version of the correlator API.

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

KUMA 4.2 introduces a beta version of correlator-ng for technical reference. Methods, algorithms, syntax of conditions, and other functions and characteristics may change before the release of the fully functional version of the correlator API.

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

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 processing functions in correlator-ng

KUMA 4.2 introduces a beta version of correlator-ng for technical reference. Methods, algorithms, syntax of conditions, and other functions and characteristics may change before the release of the fully functional version of the correlator API.

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.
`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 >, < beginning_of_substring >, [ end_of_substring ]) -> 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.

Function	Description
`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.

Function	Description
`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.

Function	Description
`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 — Router in the data center

The collector sends events to an event router in the data center, and the event router sends the events to the specified destinations: correlator and storage.

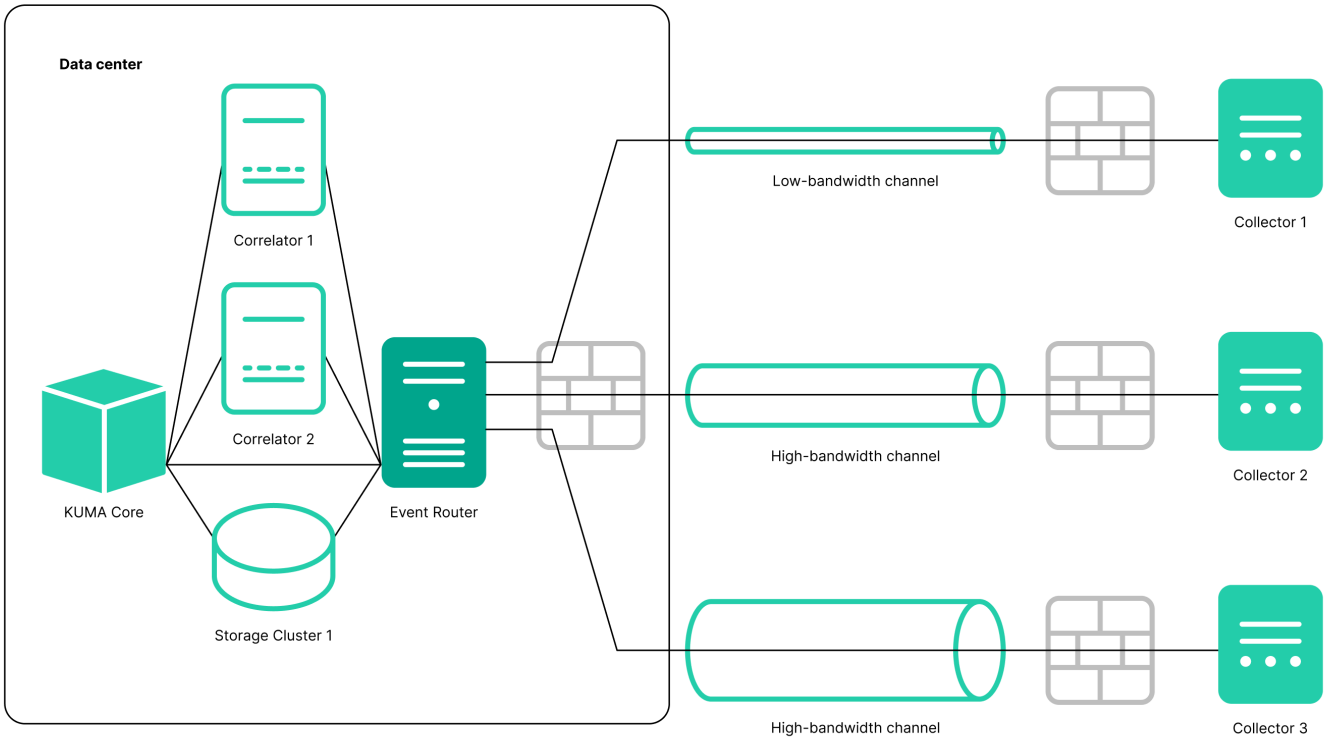


Diagram of router connection in the data center

Preconditions:

- KUMA 3.2 collectors are configured at the branch offices.
- The data center has the capacity to install an event router.
- KUMA 3.2 is installed in the data center.

Steps:

1. In the data center:
    a. Create the **Event router** service.
    b. Create **storage** and **correlator** destination points and specify them in the Event router.
    c. In the Event router on the **Advanced settings** tab, configure a filter to send events to storage and/or correlator. For example, "DeviceCustomString = correlator" or "DeviceCustomString = storage".
    d. Configure enrichment.
2. In the collectors at branch offices:
    a. Create a destination of the **eventRouter** type.
    b. Specify the URL of the event router in the data center of the branch office.
    c. If eventRouter replaces previously configured destinations, you can delete them.

Postcondition:

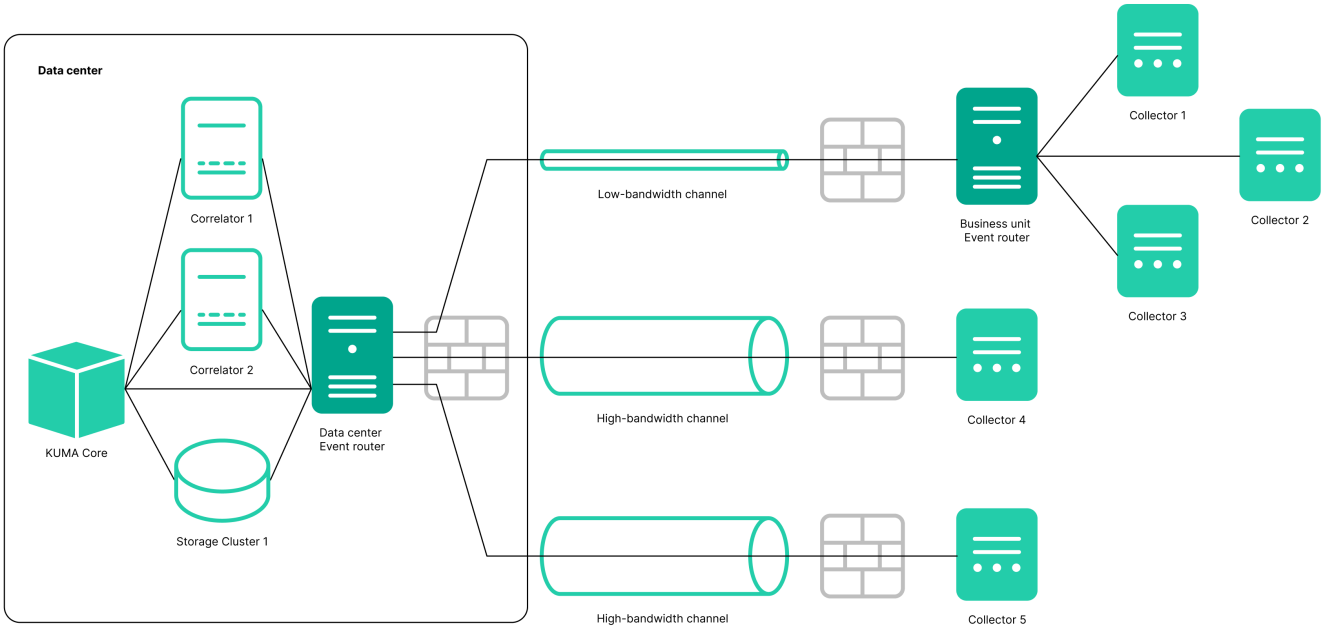
- Collectors at the branch offices are configured.
- The event router in the data center is configured.

Connections of branch offices to the data center have been optimized: for each collector, you no longer need to configure events to be sent both to storage and to the correlator in the data center. This halves the load on the link.

Routing to the storage and the correlator is performed within the data center.

Cascade connection: Multiple collectors — Router at the branch office; Router at the branch office — Router in the data center

Multiple collectors send events to the event router at the branch office, and the event router at the branch office sends events to the router in the data center, where events are then sent to the specified destinations, that is, correlators and storage.



Router cascade diagram

Preconditions:

- KUMA 3.2 collectors are configured at the branch offices.
- The data center has the capacity to install an event router.
- KUMA 3.2 is installed in the data center.

Steps:

1. In the data center:
   a. Create the **Event router** service.
   b. Create **storage** and **correlator** destination points and specify them in the Event router.
   c. In the Event router on the **Advanced settings** tab, configure a filter to send events to storage and/or correlator. For example, "DeviceCustomString = correlator" or "DeviceCustomString = storage".
2. At the branch office:
   a. Create the Event router service.
   b. Create a destination of the **eventRouter** type and specify the **URL** of the Event router in the data center.
3. In the collectors at branch offices:
   a. Create a destination of the **eventRouter** type and specify the **URL** of the Event router at the branch office.
   b. If eventRouter replaces previously configured destinations, you can delete them.

Postcondition:

1. Collectors at the branch offices are configured.
2. The event router in the data center and the event router at the branch office are configured.

The connections of branch offices with the data center are optimized: in each collector, you no longer need to configure events to be sent to the data center; it is enough to collect all events on the router and send it to the data center as one stream.

The event router must be installed on a Linux device. Only a user with the General Administrator role can create the service. You can create a service in any tenant; the tenant relation does not impose any restrictions.

You can install the event router service on a separate host or together with a collector or correlator. There is no separate section for the event router in the inventory file, so you can specify the host on which you want to install the event router in the `kuma_collector` or `kuma_correlator` section. Also, if you want to install the event router on a separate host, you need to specify this host in the inventory file in the `kuma_collector` or `kuma_correlator` section; in this way, the event router service will be upgraded as part of the next KUMA upgrade.

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:

- Create the event router service in the KUMA web interface using the Installation Wizard.
- Install the event router service on the server.

Starting the Event Router Installation Wizard

To start Event Router Installation Wizard:

1. In the KUMA web interface, in the **Resources** section, click **Event routers**.
2. This opens the **Event routers** window; in that window, click **Add**.

Follow the instructions of the installation 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 general settings of the event router:

1. On the **Basic settings** tab, fill in the following fields:

   a. In the **Name** field, enter a unique name for the service you are creating. The name must contain 1 to 128 Unicode characters.

   b. 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.

   c. 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.

   d. You can optionally add up to 4000 Unicode characters describing the service in the **Description** field.

2. On the **Advanced settings** tab, fill in the following fields:

   a. If necessary, use the **Debug** toggle switch to enable logging of service operations.

   b. 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.

   c. 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.

General settings of the event router are specified. 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 these instructions.

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 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 solution components, if necessary.

2. Close the Wizard by clicking **Save**.

The service is installed in the KUMA web interface. 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.

   If you want to install the event router service on a separate server, specify the host in the inventory file in the `kuma_correlator` group and install the router service on that host. When upgrading KUMA, the installer upgrades the event router installed on the host specified in the inventory file.

2. Create the /opt/kaspersky/kuma/ directory.

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 terms and conditions of the End User License Agreement 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 solution to work correctly, ensure that the KUMA components are able to interact with other components and applications 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 web interface> --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 receives raw events from event sources, performs normalization, and sends processed events to their destinations. The maximum size of an event that can be processed by the KUMA collector is 4 MB.

If you are using the SMB license, and both the hourly average EPS and the daily average EPS allowed by the license is exceeded for a collector, the collector stops receiving events and is displayed with a red status and a notification about the EPS limit being exceeded. The user with the General Administrator role gets a notification about the EPS limit being exceeded and the collector being stopped. Every hour, the hourly average EPS value is recalculated and compared with the EPS limit in the license. If the hourly average is under the limit, the restrictions on the collector are lifted, and the collector resumes receiving and processing events.

Installing a collector involves two steps:

- Create the collector in the KUMA web interface using the Installation Wizard. In this step, you specify the general collector settings to be applied when installing the collector on the server.
- Install the collector on the network infrastructure server on which you want to receive events.

Actions in the KUMA web interface

The creation of a collector in the KUMA web interface is carried out by using the Installation Wizard. This Wizard combines the required resources into a resource set for a collector. Upon completion of the Wizard, the service itself is automatically created based on this resource set.

To create a collector in the KUMA web interface,

Start the Collector Installation Wizard:

- In the KUMA web interface, in the **Resources** section, click the **Add event source** button.
- In the KUMA web interface in the **Resources** → **Collectors** section click **Add collector** button.

As a result of completing the steps of the Wizard, a collector service is created in the KUMA web interface.

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 identifier automatically assigned to the service in the KUMA web interface, 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 web interface, 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 web interface, in the **Resources** section, click **Add event source**.
- In the KUMA web interface 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 using 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 web interface 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 general settings of the collector:

1. On the **Basic settings** tab, fill in the following fields:

   a. 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, you cannot 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.

   b. 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 the Shared tenant can be added to the service.

   c. In the **Workers** field, if necessary, specify the number of workers that the service can run simultaneously to process events in parallel. Specify a positive integer number from 0 to 999. If 0 is specified, the number of logical CPU cores is used.

   d. You can optionally add up to 256 Unicode characters describing the service in the **Description** field.

2. On the **Advanced settings** tab, fill in the following fields:

   a. If necessary, use the **Debug** toggle switch 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 computer where the collector is installed, in the /opt/kaspersky/kuma/collector/<collector ID>/log/collector directory.

   b. 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.

   c. 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.

General settings of the collector are specified. Proceed to the next step of the Installation Wizard.

Step 2. Transport

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 using 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:

   - internal
   - tcp
   - udp
   - netflow
   - sflow
   - nats-jetstream
   - kafka
   - http
   - sql
   - file
   - 1c-log
   - 1c-xml
   - diode
   - ftp
   - nfs
   - wmi
   - wec
   - etw
   - snmp
   - snmp-trap
   - kata\edr
   - vmware
   - elastic
   - office365

- dfi
- universalAPI

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**, **wec**, or **etw** connector, agents are 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, click the **+ Add event parsing** button to open the **Basic event parsing** window, and in that window, in the **Normalizer** drop-down list, select or create a normalizer. In normalizer settings, 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 you create a new normalizer in the Installation Wizard, the default normalizer is saved in the resource set for the collector. You cannot use the created normalizer in other collectors. You can select the **Save normalizer** check box to create the normalizer as a separate resource. This will make the normalizer available for selection in other collectors of the tenant.

Adding a normalizer to a resource set

To add a normalizer to a resource set:

1. Click the **+ Add event parsing** button.

   This opens the **Basic event parsing** window with the normalizer settings. By default, the **Normalization scheme** tab is selected.

2. In the **Normalizer** drop-down list, select a normalizer. You can select a normalizer that belongs to the tenant of the collector or to the Shared tenant.

   The normalizer settings are displayed.

If you want to edit the settings of the normalizer, in the **Normalizer** drop-down list, click the pencil (🖉) icon next to the name of the normalizer to open the **Edit normalizer** window, and in that window, click the dark circle. This opens the **Basic event parsing** window with the normalizer settings. If you want to edit additional parsing settings, in the **Editing normalizer** window, hover over the dark circle and click the plus symbol that appears. This opens the **Additional event parsing** with settings of the additional parsing. For more details on configuring the additional parsing of events, see the **Creating the structure of event normalization rules** section below.

3. Click **OK**.

The normalizer is added and displayed as a dark circle on the **Event parsing** tab of the Installation Wizard. You can click the dark circle to view the settings of the normalizer.

To create a new normalizer in the collector:

1. Click the **+ Add event parsing** button.

   This opens the **Basic event parsing** window with the normalizer settings. By default, the **Normalization scheme** tab is selected.

2. If you want to create the normalizer as a separate resource, select the **Save normalizer** check box. If the check box is selected, the normalizer becomes available for selection 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. Maximum length of the name: 128 Unicode characters.

4. In the **Parsing method** drop-down list, select the type of events to receive. Depending on the selected value, you can use the predefined event field matching rules or define rules manually. When you select some parsing methods, additional settings may become available that you must specify.

   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, in the **Use raw event** drop-down list, select **Yes**.

   - **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 regular expression with named capture groups following the RE2 syntax, 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, in the **Use raw event** drop-down list, select **Yes**.

  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 **Use 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 **Use 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.

- **netflow**

  This parsing method is used to process data in all supported NetFlow protocol formats: NetFlow v5, NetFlow v9, and IPFIX.

  If you select this parsing method, you can use the predefined rules for converting events to the KUMA format by clicking **Apply default mapping**. This takes into account the source fields of all NetFlow versions (NetFlow v5, NetFlow v9, and IPFIX).

  If the **netflow** parsing method is selected for the main parsing, extra normalization is not available.

  > The default mapping rules for the **netflow** 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.

- **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**

  The normalizer uses this parsing method to process data obtained by making a selection from the database.

  This parsing method becomes available only if at the Transport step of the Installation Wizard you specified a connector of the sql type.

- **normalizer**

  This parsing method allows you to reuse a previously created normalizer as the primary or extra normalizer for processing events.

  If you select this parsing method, in the **Normalizer** drop-down list, you need to select one of the previously created normalizers of the json, cef, regexp, syslog, csv, kv, xml, or sql type. Normalizers with the **normalizer** parsing method cannot be selected.

5. In the **Keep raw event** drop-down list, specify whether you want to store the original raw event in the re-created normalized event.

   - **Don't save**—do not save the raw event. This value is selected by default.
   - **Only errors**—save the raw event in the `Raw` field of the normalized event if errors occurred when parsing it. You can use this value to debug the service. In this case, an event having a non-empty `Raw` field will indicate problems.
   - **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 is configured for them (see step 8 of these instructions):

   - **No**. This value is selected by default.
   - **Yes**. The original event fields are saved in the `Extra` field of the normalized event.

   This field is not available if the **Parsing method** is set to **normalizer**.

7. If necessary, provide an example of the data you want to process to the **Event examples** field. We recommend completing this step.

   You can add multiple event examples. Every new example must be specified on a new line. At the same time, in the **Mapping** table, its parsing result will be displayed in the **Examples** column. This lets you check whether the fields of different types of incoming events are mapped correctly.

   Limitations involved in parsing event examples

   When parsing the examples of data specified in the **Event examples** field, the following parameters are not taken into account in the extra normalizer:

   - Enrichment configured in the root normalizer
   - Filters specified in the extra normalizer
   - The **Use raw event** setting
   - Conversions in the root normalizer
   - Conversions for the value of the **Field to pass into normalizer** parameter.

   This field is not available if the **Parsing method** is set to **normalizer**.

8. In the **Mapping** table, configure the mapping of raw event fields to KUMA event fields:

   a. Click **+ Add row**.

      You can add multiple table rows or delete table rows. To delete a table row, select the check box next to it and click the **Delete** button.

   b. In the **Source** column, specify the name of the raw event field that you want to map to the KUMA event field.

      For details about the field name format, refer to the Normalized event data model article. For a description of the mapping, refer to the Mapping fields of predefined normalizers article.

      If you want to create rules for modifying the fields of the original event before writing to the KUMA event fields, click the settings (⚙) icon next to the field name to open the **Conversion** window, and in that window, click **+ Add conversion**. You can reorder the rules or delete the rules. To reorder rules, use the reorder (⋮⋮) icons. To delete a rule, click the delete (✕) icon 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 for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, 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.

- Converting the IP address to the IPv4 format:
  - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.
  - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

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**.

If at the Transport step of the Installation Wizard, you specified a connector of the file type, you can pass the name or path of the file being processed by the collector to 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** connector, the new variables in the normalizer will only work with destinations of the internal type.

c. In the **KUMA field** column, select a KUMA event field from the drop-down list. You can find the KUMA event field by typing its name. If the name of the KUMA event field begins with `DeviceCustom*` or `Flex*`, enter a unique custom label in the **Label** field, if necessary.

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, you can configure a mapping for **SourceAddress** and **SourceHostName**, or **DestinationAddress** and **DestinationHostName** KUMA event fields. As a result of enrichment, the event card includes a **SourceAssetID** or **DestinationAssetID** KUMA event field, and a link to the asset card. As a result of enrichment, asset information is also 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.

The **Mapping** table is not available if the **Parsing method** is set to **normalizer**.

9. Click **OK**.

The normalizer is created and displayed as a dark circle on the **Event parsing** tab of the Installation Wizard. You can click the dark circle to view the settings of the normalizer. If you want to edit additional parsing settings, hover over the dark circle and click the plus symbol that appears. This opens the **Additional event parsing** with

settings of the additional parsing. For more details on configuring the additional parsing of events, see the **Creating the structure of event normalization rules** section below.

When displaying a normalization scheme, the structure of nested normalizers is not displayed. To view this structure, you need to go to the normalizer card by clicking its name link. If you are using a nested normalizer with extra normalization, keep in mind that the structure of the output data may change. In that case, it is important to understand which field is passed to the other extra normalizer.

Enriching normalized events with additional data

You can create enrichment rules in the normalizer to add extra data to created normalized events. Enrichment rules are stored in the normalizer in which they were created.

The **Enrichment** tab is not available if on the **Normalization scheme** tab, the **Parsing method** is set to **normalizer**.

To add enrichment rules to the normalizer:

1. Select the main or additional normalization rule to open a window, and in that window, select the **Enrichment** tab.

2. Click the **+ Add enrichment** button.

   The enrichment rule settings section is displayed. You can add multiple enrichment rules or delete enrichment rules. To delete an enrichment rule, click the delete (✕) icon next to it.

3. In the **Source kind** drop-down list, select the type of the enrichment source. When you select some enrichment source types, additional settings may become available that you must specify.

   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 enrichment fields** setting 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 extended schema array 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. 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.
Mapping	Event fields for data transfer: • **Dictionary field** specifies dictionary fields from which data is to be transmitted. The available fields depend on the selected dictionary resource. • **KUMA field** specifies event fields to which data is to be transmitted. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written there.

  The first field in the table (**Dictionary field**) is taken as the key with which the fields selected from the event as key fields are matched (**KUMA field**). As the key in the **Dictionary field**, you must select an indicator of compromise by which the enrichment is to be performed, for example, IP address, URL, or hash. In the rule, you must select the event field that corresponds to the selected indicator of compromise in the dictionary field.

  If you want to select multiple key fields, you can specify them using `|` as a separator (when specifying in the web interface or importing as a CSV file), for example, `<IP address>|<user name>`.

  You can add new table rows or delete table rows. To add a new table row, click **Add new element**. To delete a row in the table, click 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 for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, 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.

- Converting the IP address to the IPv4 format:

  - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.
  - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

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**.

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.

  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 the result of 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. You cannot select a value if in the **Source kind** drop-down list, you selected **table**.

5. If you want to enable details in the normalizer log, enable the **Debug** toggle switch. The toggle switch is turned off by default.

6. Click **OK**.

The event enrichment rules with additional data are added to the normalizer into the selected parsing rule.

Configuring parsing linked to IPv4 addresses

If at the Transport step of the Installation Wizard, you specified a connector of the udp, tcp, or http type, you can forward events from multiple IPv4 addresses from sources of different types to the same collector, and have the collector apply the specified normalizers. To do this, you need to specify several IPv4 addresses and select the normalizer that you want to apply to events coming from the specified IPv4 addresses. The following types of normalizers are available: **json**, **cef**, **regexp**, **syslog**, **csv**, **kv**, **xml**, **normalizer**.
  ```

If you select a connector type other than **udp**, **tcp**, or **http** in a collector with configured normalizers and linking to IPv4 addresses, the **Parsing settings** tab is no longer displayed, and only the first of the previously specified normalizers is specified on the **Event parsing** tab of the Installation Wizard. The **Parsing settings** tab is hidden immediately, and the changes are applied after the resource is saved. If you want to restore the previous settings, exit the Installation Wizard without saving.

For normalizers of the **Syslog** and **regexp** types, you can use of a chain of normalizers. In this case, you can specify additional 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 IPv4 addresses:

1. Select the **Parsing settings** tab and click the **+ Event source** button.

   A group of parsing settings is displayed. You can add multiple parsings or delete parsings. To remove a parsing, click the delete (✕) icon next to it.

2. In the **IP address (-es)** field, enter the IPv4 address from which events will be sent. You can specify multiple IP addresses separated by commas. The length of the IPv4 address list is unlimited; however, we recommend limiting the number of IPv4 addresses to keep the load on the collector balanced. You must specify a value in this field if you want to apply multiple normalizers in one collector.

   > The IPv4 address must be unique for each normalizer. KUMA checks the uniqueness of IPv4 addresses, and if you specify the same IPv4 address for different normalizers, the `Field must be unique` message is displayed.

   If you want to send all events to the same normalizer without specifying IPv4 addresses, we recommend creating a separate collector. To improve performance, we recommend creating a separate collector with one normalizer if you want to apply the same normalizer to events from a large number of IPv4 addresses.

3. In the **Normalizer** drop-down list, create or select a normalizer. You can click the arrow icon next to the drop-down list to select the **Parsing flowcharts** tab.

   Normalization will be triggered if at the Transport step of the Installation Wizard, you specified a connector of the udp, tcp, or http type. For a **http** connector, you must specify the header of the event source. Taking into account the available connectors, the following normalizer types are available for automatic source recognition: **json**, **cef**, **regexp**, **syslog**, **csv**, **kv**, and **xml**.

4. If you selected a normalizer of the **Syslog** or **regexp** type, and you want to add a conditional normalization, click the **+ Add conditional normalization** button. Conditional normalization is available if in the **Mapping** table of the main normalizer, you configured the mapping of the source event field to the **DeviceProcessName** KUMA event field. Under **Condition**, in the **DeviceProcessName** field, specify a process name, and in the drop-down list, create or select a normalizer. You can specify multiple combinations of the **DeviceProcessName** KUMA event field and a normalizer. Normalization is performed until the first match.

Parsing with linking to IPv4 addresses is configured.

Creating a structure of event normalization rules

To implement complex event processing logic, you can add multiple event parsing rules to the normalizer. Events are switched between the parsing rules depending on the specified conditions. Events are processed sequentially in accordance with the creation order of the parsing rules. The event processing path is displayed as arrows.

To create an additional parsing rule:

1. Create a normalizer. For more information on creating normalizers, see the **Adding a normalizer to a resource set** section earlier in this article.

   The created normalizer is displayed as a dark circle on the **Event parsing** tab of the Installation Wizard.

2. Hover over the dark circle and click the plus sign that appears.

3. In the **Additional event parsing** window that opens, configure the additional event parsing rule:

   - **Extra normalization conditions** tab:

     If you want to send a raw event for extra normalization, in the **Use raw event** drop-down list, select **Yes**. 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, in the **Use raw event** drop-down list, select **Yes**.

     To send only the events with a specific field to the additional normalizer, specify the field in the **Field to pass into normalizer** field.

     On this tab, you can define other conditions that must be satisfied for the event to be sent for additional parsing.

   - **Normalization scheme** tab:

     On this tab, you can configure event processing rules, similar to the main normalizer settings. The **Keep raw event** setting is not available. The **Event examples** field displays the values specified when the normalizer was created.

   - **Enrichment** tab:

     On this tab, you can configure enrichment rules for events. For more details on configuring enrichment rules, see the **Enriching normalized events with additional data** section earlier in this article. This tab is not available if on the **Normalization scheme** tab, the **Parsing method** is set to **normalizer**.

4. Click **OK**.

An additional parsing rule is added to the normalizer and displayed as a dark box. The dark box specifies the conditions that trigger the additional parsing rule.

You can do the following:

- You can chick the additional parsing rule to edit its settings.
- You can find an additional parsing rule by entering its name in the field in the upper part of the window.
- Create a new additional parsing rule. To do this, hover over the additional parsing rule and click on the plus icon that appears.
- Duplicate an additional parsing rule. To do this, hover over the additional parsing rule and click on the duplicate ( ) icon.
- Delete an additional parsing rule. To do this, hover over the additional parsing rule and click on the trash can icon.

- Change the position and order of additional normalizers. To do this, select the check box next to the extra normalizer and in the toolbar, click **Move**. This opens the **Move normalizer** window; in that window, select the required values in the **New parent** and **Place after** drop-down lists and click **Save**. If nothing is selected in the **Place after** drop-down list, the additional normalizer is moved to the beginning of the sequence of additional normalizers in the selected parent normalizer. In the lower part of the additional normalizer section, the number of additional normalizers associated with it is indicated.

  You can move multiple additional normalizers of the same level together. If you move an additional normalizer that has child normalizers, the whole chain is moved.

- Adjust the display of the structure of normalization rules in the toolbar using the **Collapse to default**, **Expand all**, **Compact view** buttons.

  By default, all additional normalizers at second level and below are collapsed. The main normalizer is the zero level, followed by the first level of extra normalizers, and the rest of the extra normalizers are collapsed. When you go to the normalization scheme, the scheme is displayed collapsed by default. For each item in the normalizer tree, you can collapse and expand child items using the **Collapse** ( ) and **Expand** ( ) icons. The icons are displayed when you hover over the parent normalizer.

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.

When configuring filters, we recommend to adhere to the chosen normalization scheme. In filters, use only KUMA service fields and the fields that you specified in the normalizer in the **Mapping** and **Enrichment** sections. For example, if the DeviceAddress field is not used in normalization, avoid using the DeviceAddress field in a filter because such filtering will not work.

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:

   - You can click the **Add condition** button 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 **op** 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**, **InActiveDirectoryGroup** operators are selected. This check box is cleared by default.

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**—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 are displayed, which you can use to precisely define the value to 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 using the × button.

- The **Add group** button is used to add groups of conditions. The **AND** can be switched between **And**, **OR**, **NOT**.

  A condition group can be deleted using 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 using 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 aggregation rules 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, use the **Add field** button to select the fields that will be used to identify the same types of events. You can select the following service fields: SourceAssetID, DestinationAssetID, DeviceAssetID, SourceAccountID, DestinationAccountID. Asset IDs are added after event normalization, which means you can use them to aggregate events. The rest of the service fields are not used in aggregation. Selected events can be deleted using 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 using the buttons with a cross icon.

7. In the **Sum fields** section, you can use the **Add field** button to select the fields whose values will be summed during the aggregation process. Selected events can be deleted using 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

   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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

Aggregation rule added. You can delete it using 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.

Incoming events are first enriched with data from the sources that you select at this step of the wizard, and then the events are automatically enriched with asset information. For example, if you have dns enrichment configured, which leads to the IP address in the event being enriched with the corresponding DNS name of an asset, the event is enriched with this asset.

The collector receives information about assets from the KUMA Core and enriches events with this information as follows:

- If the name field of the event (for example, `SourceHostName`) contains an FQDN that matches the FQDN of an asset, the collector adds a `SourceAssetID` field with the ID of thif asset. This does not take into account the value of the `SourceAddress` field.
- If an IP address field of an event (for example, `DestinationAddress`) includes the IP address of an asset, and the name field is empty, the collector adds a `DestinationAssetID` field with the ID of the asset that has the matching IP address.
- If a name field of an event does not match any FQDN from the asset database, the collector does not check the IP address field or add an asset ID. This applies to cases in which the `HostName` field contains an FQDN that does not match any asset, as well as cases in which an incomplete name is specified in the `HostName` field (for example, only the host name without a domain).

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 collector resources,

1. Click **Add enrichment**.

   This opens the enrichment rules settings block.

2. In the **Enrichment rule** drop-down list, select an enrichment rule.

The enrichment rule is added to the resource set for the collector.

To create a new enrichment rule in a resource set of the collector:

1. Click **Add enrichment**.

   This opens the enrichment rules settings block.

2. In the **Name** field, enter a unique name for the rule. The name must contain 1 to 128 Unicode characters.

3. In the **Enrichment rule** drop-down list, select **Create new**.

4. In the **Source kind** drop-down list, select the source of data for enrichment and define its corresponding settings:

- **template**

  This type of enrichment is used when you need to write the result of 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}}
  ```

- **table**

  This type of enrichment is used if you need to add a value from the dictionary of the **Table** 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.
Mapping	Event fields for data transfer: • **Dictionary field** specifies dictionary fields from which data is to be transmitted. The available fields depend on the selected dictionary resource. • **KUMA field** specifies event fields to which data is to be transmitted. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written there.

  The first field in the table (**Dictionary field**) is taken as the key with which the fields selected from the event as key fields are matched (**KUMA field**). As the key in the **Dictionary field**, you must select an indicator of compromise by which the enrichment is to be performed, for example, IP address, URL, or hash. In the rule, you must select the event field that corresponds to the selected indicator of compromise in the dictionary field.

  If you want to select multiple key fields, you can specify them using `|` as a separator (when specifying in the web interface or importing as a CSV file), for example, `<IP address>|<user name>`.
  ```

You can add new table rows or delete table rows. To add a new table row, click **Add new element**. To delete a row in the table, click the ✕ button.

- **enrichment by assets**

  You can use this type of enrichment when you need to add enrichment of events with assets to specific collectors.

  To enrich events with assets under **Mapping**, fill in the mapping table of the KUMA event fields and the fields used to record values from the hash table:

  1. Click **Add**.

  2. In the **KUMA field** drop-down list, select one or more event fields that you want to map to values from the hash table and write to the corresponding asset information field in the table.

  3. In the **KUMA event field to write to** drop-down list, select the field into which you want to write the event ID, if the event field (**KUMA field**) matches the value from the hash table.

     Possible parameter values in the field mapping table

KUMA field	KUMA event field to write to

`DestinationAddress`
`DestinationHostName`
`DeviceAddress`
`DeviceHostName`
`SourceAddress`
`SourceHostName`
`DestinationMacAddress`
`DeviceMacAddress`
`SourceMacAddress`
`DeviceCustomString1`
`DeviceCustomString2`
`DeviceCustomString3`
`DeviceCustomString4`
`DeviceCustomString5`
`DeviceCustomString6`
	`SourceAssetID`
	`DeviceAssetID`
	`DestinationAssetID`

     If the user selects multiple fields, the **AND** operator is applied to field values. If the user adds multiple strings in the **KUMA field** column, the **OR** operator is applied to field values. The matching begins from the first field in the list. If the value of the field matches the value of the attribute, matching stops.

     For example, suppose that the following data is received in an event:

     ```
     SourceAddress=192.168.x.xx
     SourceHostName=kuma.example.com.
     ```

     There are two assets in KUMA: *company.example.com* (IP address: 192.168.x.xx, FQDN: company.example.com) and *kuma.example.com* (IP address: 192.168.y.yyy, FQDN: kuma.example.com). Matching is performed as follows:

     - If the user separately specified `sourceAddress` → `sourceAssetID`, and then `sourceHostname` → `sourceAssetID`, then the asset matches the first condition (company.example.com).

     - If it is more important for the user to get the `sourceHostname` value (kuma.example.com), then `sourceHostname` → `sourceAssetID` must be specified first, followed by `sourceAddress` → `sourceAssetID`.

     - If the user specifies `sourceAddress` + `sourceHostname` → `sourceAssetID`, enrichment does not occur.

At least one matching condition must be specified in the field mapping table.

You can also click the **Apply default mapping** button above the table. In this case, the values of **KUMA field** and **KUMA event field to write to** are applied.

Default values of parameters in the field mapping table

KUMA field	KUMA event field to write to
DestinationAddress	DestinationAssetID
DestinationHostName	DestinationAssetID
DeviceAddress	DeviceAssetID
DeviceHostName	DeviceAssetID
SourceAddress	SourceAssetID
SourceHostName	SourceAssetID

- **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 enrichment fields** setting 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 extended schema array 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.
  - Under **Conversion**, 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 for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, 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.

- Converting the IP address to the IPv4 format:

  - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.

  - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

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**.

- **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** is the number of workers that the service can run simultaneously to process events in parallel. Specify a positive integer number from 0 to 999. If 0 is specified, the number of logical CPU cores is used.
- **Max tasks**—maximum number of simultaneously fulfilled requests. By default, this value is equal to the number of vCPUs of the KUMA Core server.
- **Cache TTL**—the lifetime (in seconds) of the values stored in the cache. The default value is `60` (seconds).
- **Cache disabled**—you can use this toggle switch to enable or disable caching. Caching is enabled by default.
- **Cache update disabled**—if cache update is enabled, then after the cache TTL expires, the collector sends repeat queries to the DNS server to update the information stored in the cache. This setting is available only when cache is enabled. By default, cache update is enabled.
- The **Recursive queries** setting is available starting with KUMA 3.4.1. You can use this toggle switch to make a KUMA collector send recursive queries to authoritative DNS servers for the purposes of enrichment. The default value is **Disabled**.

- **cybertrace**

  For systems with a high load on the collector or correlator, we recommend using the cybertrace-http enrichment type. This type returns a structured response in JSON format and correctly distinguishes between strings and arrays of custom attributes, preventing parsing errors.

  When enriched through cybertrace, the collector gets an unstructured response. If the value of a custom attribute in CyberTrace contains square brackets, for example, **[str1**, **str2]**, KUMA treats it as an array, not a string.

  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.

- **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 is a new streaming event enrichment type in CyberTrace that allows you to send a large number of events with a single request to the CyberTrace API. Recommended for systems with a lot of events. The performance of cybertrace-http is superior to that of the cybertrace type.

  Limitations:

  - The cybertrace-http enrichment type cannot be used for retroscan in KUMA.
  - If the cybertrace-http enrichment type is being used, detections are not saved in CyberTrace history in the Detections window.

  Available settings:

  - **Name** (required) is a unique name for the enrichment rule. Up to 128 Unicode characters.
  - **Tenant** (required) is the name of the tenant in which you plan to work with CyberTrace.
  - **URL** (required)—in this field, you can specify the URL of a CyberTrace server to which you want to send requests and the port that CyberTrace API is using. The default port is 443.
  - **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 1,000,000. 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.

- **Major version of CyberTrace** selects 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**.

- The **Debug** toggle switch enables resource logging. The toggle switch is turned off by default.

- **Tags** for resource search.

- **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, under **Mapping geographic data to event fields**, 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 pre-configured 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.

- **check DLL Hijacking**

  DLL Hijacking is an attack technique that involves delivering vulnerable legitimate software along with a malicious dynamic link library (DLL) to the target system. 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 status, KUMA sends a request to KSN with information from event fields. In response to the request, KSN sends the analysis result.

  You can configure DLL Hijacking attack detection in the collector or correlator.

  To use enrichment such as **check DLL Hijacking**, your license must include the AI module. The General administrator must also accept the additional KSN Statement in the **Kaspersky Security Network** section (**Settings** → **Integrations** → **AI services** → **DLL Hijacking detection**).

Available settings:

- **Workers** is the number of workers that the service can run simultaneously to process events in parallel. Specify a positive integer number from 0 to 999. If 0 is specified, the number of logical CPU cores is used.

- **Maximum cache size** is the maximum cache size for KSN responses, in megabytes. The default value is `128 MB`.

  Each KSN response has its own lifetime, during which it is retained in the cache. If KUMA sends a second request for a DLL file for which KSN has already issued a response, and the lifetime of that response has not yet expired, the request is not actually sent to KSN, but the response is taken from the cache.

- If the **Maximum number of events in enrichment queue** is exceeded, new events are not processed. The default value is `1,000,000`. `0` means that event processing stops if the queue contains any number of events.

- **Perform a repeat request**. KSN may return the **Unclassified** status. This status means that KSN has accepted the request and is currently processing it, but there is no status yet. Processing may take a long time, so the request must be re-sent. If this setting is enabled, specify additional settings:

  - If the **Create duplicates** setting is enabled, KUMA creates a duplicate event and uses it for repeated requests to KSN. In this case, the main event is routed in accordance with the workflow. A duplicate event is written to the database when one of the following conditions is satisfied:

    - A status other than **Unclassified** is received from KSN.
    - The disk buffer configured using the **Buffer size** setting is full.
    - The **Request count** is exceeded.

    If this setting is disabled, KUMA continues to send requests using the main event. The event is written to the database when one of the following conditions is satisfied:

    - A status other than **Unclassified** is received from KSN.
    - The disk buffer configured using the **Buffer size** setting is full.
    - The **Request count** is exceeded.

  - The **Request count** is the maximum number of requests to KSN. If, after the specified number of requests, a status other than **Unclassified** is not received, the event is written to the database with the **Unclassified** status. The default value is `10`.

  - **Poll interval** is the interval between requests, in minutes. The default value is `1 minute`. The minimum value is `1 minute`.

  - **Buffer size** is the size of the disk buffer for events awaiting a repeat request, in megabytes. The default value is `10240 MB`. The minimum value is `10 MB`.

- **Proxy** specifies proxy server settings, if one is needed to connect to the KSN server. You can select one of the available proxy servers or create a new proxy server. By default, a proxy server is not used.

- **Mapping** is a table of parameters that must be sent in a KSN request to get a status, and a list of the corresponding KUMA event fields. List of parameters in the request:

  - **dll hash (md5)**. Recommended field for mapping: OldFileHash.
  - **dll path**. Full path including the file name; recommended field for mapping: OldFilePath. The path may contain personal data, so KUMA sends this parameter to KSN in an anonymized form. For example:

    Path before anonymization: `C:\Windows\Device\HarddiskVolume1\Users\Alex\malware.dll`

    The same path in anonymized form: `device\harddiskvolume1\users\username`
  - **process hash (md5)**. Recommended field for mapping: FileHash.
  - **process path**. Full path including the file name. The path may contain personal data, so KUMA sends this parameter to KSN in an anonymized form. For example:

    Path before anonymization: `C:\Windows\Device\HarddiskVolume1\Users\Alex\malware.dll`

    The same path in anonymized form: `device\harddiskvolume1\users\username`
  - **dll name**. Recommended field for mapping: OldFileName.
  - **process name**. Recommended field for mapping: FileName.
  - **process signature**. Optional; recommended field for mapping: FileID. In this field, you need to specify the KUMA field that passes the Organization from the Subject of the signature. If these values are not available, leave the signature empty. To add this parameter, click the **Add** button, then in the **DLL Hijacking field** field, select **process signature** from the drop-down list.
  - **dll size**. Optional; recommended field for mapping: OldFileSize. To add this parameter, click the **Add** button, then in the **DLL Hijacking field** field, select **dll size** from the drop-down list.

  For each parameter, specify a corresponding KUMA event field in the table.

5. If necessary, use the **Debug** toggle switch to enable logging of service operations.
6. If necessary, in the **Tags** drop-down list, select thetags for the enrichment rule.
7. If required, add a description for the enrichment rule in the **Description** field. The rule description can contain up to 4000 Unicode characters.
8. Under **Filter parameters**, 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.

   Adding conditions and condition groups for the filter

   To adding conditions and condition groups for the filter:

   1. When creating an enrichment rule, make sure that under **Filter parameters**, **Builder** is selected.
   2. Click the **Add condition** button and in the drop-down lists that appear one by one, select event values and the filter operator.

      You can add multiple conditions to a filter.
   3. If necessary, click the **Add group** button to add a group of conditions for the filter.

      You can add multiple condition groups.
   4. Click a logical operator (**AND**, **OR**, **NOT**). **AND** is selected by default.

The new enrichment rule was added to the resource set for the collector.

LDAP enrichment

To enable enrichment using LDAP:

1. Click **Add enrichment with LDAP data**.

   This opens the group of settings for LDAP enrichment.

2. In the **LDAP accounts mapping** field, add one or more LDAP connections created in the LDAP server integration settings.

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 whose data should be compared to an LDAP attribute.

   - In the **LDAP attribute to receive** column, specify the attribute that must be compared to the KUMA event field. The drop-down list contains some of the standard attributes that can be used for enrichment. You can add **custom attributes** to the list.

     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.

        Default 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.

        List of attributes imported from AD

        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 attributes** drop-down list in the collector will be automatically appended with 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, delete the row containing the attribute from the collector's mapping table manually. 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.

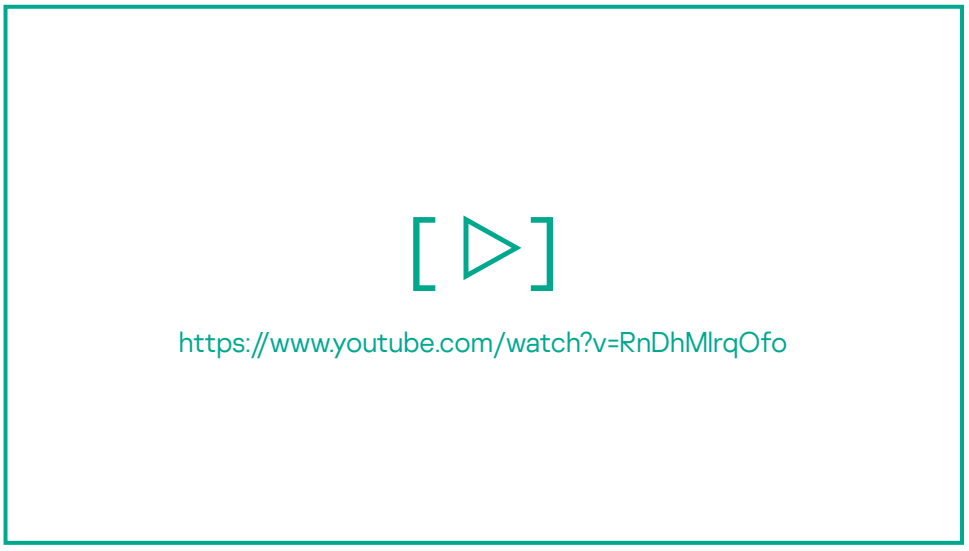
- In the **Event field to write to** column, specify the KUMA event field where the ID of the user account imported from LDAP needs to be placed in case of successful mapping.

You can use the **Add** button to add a row to the table, and remove one using the **Delete** button. You can use the **Apply default mapping** button to fill the mapping table with standard values.

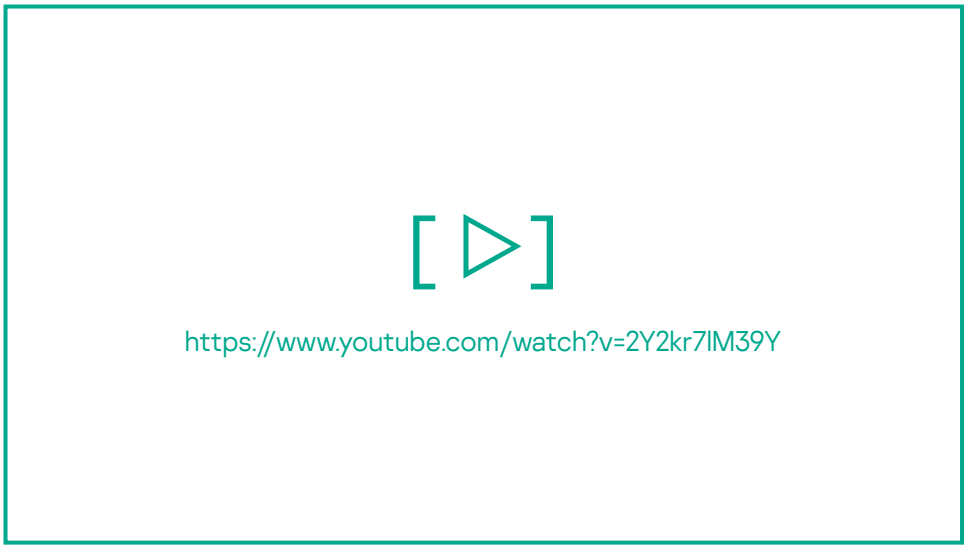
A mapping is added to the resource list of the collector for enriching events with data received from LDAP.

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.



https://www.youtube.com/watch?v=RnDhMlrqOfo



https://www.youtube.com/watch?v=2Y2kr7lM39Y

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. If necessary, events can be sent elsewhere, for example, to the event router. In that case, select the 'internal' connector at the Transport step. There can be more than one destination point.

To add an existing destination to a collector resource set:

1. In the **Routing** step of the Installation Wizard, click **Add**.
2. This opens the **Create destination** window; in that window, select the type of destination you want to add.
3. 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 .
4. 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 **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:

   - internal
   - nats-jetstream
   - tcp
   - http
   - diode
   - 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 these instructions.

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 resource set 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 use the **Save and restart services** and **Save and update service configurations** buttons.

  A resource set 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 solution. 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 resource set 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 the service must be installed on the server intended for receiving events.

If a wmi, wec, or etw 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 in the KUMA web interface, 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. 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 web interface> --api.port <port used for communication with the installed component>`

   Example: `sudo /opt/kaspersky/kuma/kuma collector --core https://kuma.example.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 web interface 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. 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 are automatically specified in the command.

   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.

3. Run the command again by adding the `--install` key:

```
sudo /opt/kaspersky/kuma/kuma collector --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 web interface> --api.port <port used for communication with the installed component> --install
```

Example: `sudo /opt/kaspersky/kuma/kuma collector --core https://example.com:7210 --id XXXX --api.port YYYY --install`

4. Add KUMA collector port to firewall exclusions.

> For the solution to work correctly, ensure that the KUMA components are able to interact with other components and applications 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 send it for processing.

Validating collector installation

To verify that the collector is ready to receive events:

1. In the KUMA web interface, open **Resources** → **Active services**.
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 web interface:*

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 in the in the **Resources** section of the KUMA web interface, in the **Keep raw event** drop-down list for the **Normalizer** resource, you have selected **Only errors**.
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 KUMA Collectors 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 collectors that use WEC, WMI, or ETW connectors as the transport, make sure that a unique port is used for connecting to the agent. 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 (refer to 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 the event stream using nginx, you need to create and configure an nginx 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.
```

With a large number of active services and users, you may need to increase the limit of open files in the nginx.conf settings. For example:

```
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 nginx 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 KUMA 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 web interface, 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 the agent in the KUMA web interface**
2. **Creating an agent service in the KUMA web interface**
3. **Installing the server portion of the agent to the asset that will send events**

KUMA agents for Windows devices can be created as follows:

- Manually by creating a set of agent resources and creating an agent service in the KUMA web interface.
- Automatically when creating a collector of the wmi, wec, or etw 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 send an event. In a manually created agent, a connector of the etw type can be used in only one connection. If you configure multiple connections in a manually created etw agent, the status of the etw agent status is green, but events are not transmitted and an error is logged in the etw agent log.

Multiple agents can be installed on a device; the version of all such agents must be the same.

If an older version of the agent is already installed on the device where you want to create an agent, you must first stop the installed agent (remove the agent from a Windows device or restart the agent service on a Linux device), and then you can proceed to create a new agent. However, if the version of the installed agents is the same as the version of the agent that you want to create, stopping the agents is not necessary.

When creating and running an agent whose version is 3.0.1 or later, you must accept the terms and conditions of the End User License Agreement.

Creating a set of resources for an agent

In the KUMA web interface, an agent service is created based on the resource set for an agent that unites connectors and destinations. 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.

To create a resource set for an agent in the KUMA web interface:

1. In the KUMA web interface, under **Resources** → **Agents**, click **Create new**.

   This opens the agent creation window. The left part of the window displays tabs with base settings of the agent and connections. The **Base settings** tab is active.

2. On the **Base settings** tab:

   - 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 storage.
   - If you want to enable logging of service operations, enable the **Debug** toggle switch.
   - If you want to view the names of services or addresses of hosts from which the event came, enable the **Track event route** toggle switch.

     The **Track event route** toggle switch is available if at least one **internal** destination is specified in the connections. By default, the toggle switch is **Disabled**.
   - You can optionally add up to 256 Unicode characters describing the service in the **Description** field.

3. Go to the tab of an existing connection or create a connection by clicking the **Add** button in the lower part of the left pane, then go to the tab of the newly created connection to edit its settings.

   By default, a connection named **Config #1** is created for a new agent. The name of the connection follows the **Config #<number>** pattern.

   You can create multiple connections for an agent. If necessary, you can manage connections:

   - Rename connections
   - Duplicate connections
   - Delete connections

4. If necessary, in the **Name of connection** field, rename the connection for your convenience when managing it, for example, to be able to figure out from which connection and from which agent events arrived. If you leave the field blank, a name is assigned following the **Config #<number>** pattern.

The name can contain from 1 to 128 characters. The name can contain only letters and numerals and cannot contain special characters. Leading and trailing spaces are removed. When pasting a name into the field from the clipboard, if the text contains a newline, paragraph, or indentation, these characters are replaced with a space. You can reuse a name for multiple connections within the same agent.

If you have enabled event route tracing, then when viewing event information, the **Events** section displays the name of the connection from which the event was received.

5. 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
       - snmp-trap
       - etw
       - kata/edr
       - office365
       - dfi
       - vmware

- 1c-log
- 1c-xml

When using the **tcp**, **udp**, **http**, **wec**, **wmi**, **etw** connector type, at the normalization stage, IP addresses of the assets from which the events were received are written to the DeviceAddress event field if it is empty.

The ability to edit previously created wec, wmi, or etw connections in agents, collectors, and connectors is limited. You can change the connection type from **wec** to **wmi** or **etw** and back, but you cannot change the **wec**, **wmi**, or **etw** connection to any other connection type. When editing other connection types, you cannot select the **wec**, **wmi**, or **etw** types. You can create connections without any restrictions on the types of connectors.

When adding an (existing or new) wmi, wec, or etw connector for an agent, the **TLS mode** and **Compression** settings are not displayed on the agent, but the values of these settings are stored in the agent's configuration. For a new connector, these settings are disabled by default.
If TLS mode is enabled for an existing connector that is selected in the list, you cannot download the agent configuration file. In this case, to download the configuration file, you must go to the connector resource that is being used on the agent and disable TLS mode.

The connector is added to the selected connection of the agent's resource set. The created connector is only available in this resource set and is not displayed in the web interface **Resources → Connectors section**.

6. 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.
  - Enable or disable the **State** toggle switch to enable or disable the sending of events to the destination. This toggle switch is turned on by default.

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.

7. Repeat steps 3–5 for each agent connection that you want to create.

8. Click **Save**.

The resource set for the agent is created and displayed under **Resources** → **Agents**. Now you can create an agent service in KUMA.

Managing connections for an agent

You can manage the connections created for the agent by renaming, duplicating, or deleting them.

Renaming a connection

By default, the name of a new connection created in a resource set for an agent follows the **Connection <number>** pattern. For your convenience when managing connections, you can rename them, for example, to make it clear at a glance from which connection and from which agent an event was received.

To rename a connection:

1. In the KUMA web interface, in the **Resources** → **Agents** section, create a new agent or click an existing agent in the table.

2. If necessary, create a connection for the agent by clicking the **Add** button in the lower part of the left pane.

3. Go to the tab of the connection that you want to rename.

4. In the **Connection name** field, edit the name of the connection.

   The name can contain from 1 to 128 characters. The name can contain only letters and numerals and cannot contain special characters. Leading and trailing spaces are removed. When pasting a name into the field from the clipboard, if the text contains a newline, paragraph, or indentation, these characters are replaced with a space. You can reuse a name for multiple connections within the same agent.

5. Click **Save**.

The connection is renamed. If you have enabled event route tracing, then when viewing event information, the **Events** section displays the name of the connection from which the event was received.

Duplicating a connection

If you want to create a connection for an agent based on an existing connection, you can create a copy of the connection with identical settings.

To duplicate a connection:

1. In the KUMA web interface, in the **Resources** → **Agents** section, create a new agent or click an existing agent in the table.
2. If necessary, create a connection for the agent by clicking the **Add** button in the lower part of the left pane.
3. Go to the tab of the connection that you want to duplicate and click the **Duplicate** button.

   A connection is created for the agent with the same settings as the original connection. The new connection is created with one of the following names:

- If the original connection name followed the **Connection <number>** template, the duplicated connection name also follows this template. The connection is named "Connection <number+1>", where <number> is the number of the last created connection whose name followed the same template.
- If the original connection had been renamed, the new connection gets the same name. You can reuse a name for multiple connections within the same agent.

You can edit the name of the new connection in the **Name of connection** field on the tab of the connection.

Removing a connection

If the agent has more than one connection, you can delete a connection.

To delete a connection:

1. In the KUMA web interface, in the **Resources** → **Agents** section, create a new agent or click an existing agent in the table.
2. If necessary, create a connection for the agent by clicking the **Add** button in the lower part of the left pane.
3. Go to the tab of the connection that you want to delete and click the **Delete** button.

   You cannot restore a deleted connection, but you can recover the version of the agent in which this connection was previously saved through the change history.

Creating an agent service in the KUMA web interface

When a resource set is created for an agent, you can proceed to create an agent service in KUMA.

To create an agent service in the KUMA web interface:

1. In the KUMA web interface, under **Resources** → **Active services**, click **Add service**.
2. In the opened **Choose a service** window, select the resource set that was just created for the agent and click **Create service**.

   The agent service is created in the KUMA web interface 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.

Automatically created agents

When creating a collector with wec, wmi, or etw 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.
- If in a connector of the etw type, the session name is specified incorrectly, the wrong provider is specified in the session, or an incorrect method is specified for sending events (to send events correctly, on the Windows Server side, you must specify "Real time" or "File and Real time" mode), events will not arrive from the agent, an error will be recorded in the agent log on Windows, and the status of the agent will be green. At the same time, no attempt will be made to get events every 60 seconds. If you modify session settings on the Windows side, you must restart the etw agent and/or the session for the changes to take effect.

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 send a message.

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.

Multiple agents can be installed on a device; the version of all such agents must be the same.

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. If you do not want to start KUMA agents manually, we recommend installing agents using an application that automatically starts applications whenever the server is restarted, for example, the Supervisor application. If you want to start KUMA agents automatically, specify the automatic start and restart settings in the KUMA configuration file. For more information on configuring automatic starting and restarting, see the official documentation of applications for automatically starting applications.

Example configuration in Supervisor

```
[program:agent_<agent_name>] command=sudo ==/opt/kaspersky/kuma/agent agent --core https://<FQDN>:<port>
autostart=true
autorestart=true
```

where:

- <agent_name> is the name of the KUMA agent.
- <FQDN> is the fully qualified domain name of the KUMA Core server.
- <port> is the port number of the KUMA Core.

To install a KUMA agent to a Linux asset:

1. On the Linux device on which you want to install the KUMA agent, create directories for the KUMA configuration file and agents, for example:
   - /opt/kaspersky/kuma
   - /opt/kaspersky/agent
2. Place the KUMA configuration file in the directory created for it. The KUMA configuration file can be found inside the installer in the /kuma-ansible-installer/roles/kuma/files directory.

   Make sure the kuma file has sufficient rights to run.
3. Create the KUMA user:

   ```
   sudo useradd --system kuma && usermod -s /usr/bin/false kuma
   ```

4. Grant the KUMA user access to the directory with the KUMA configuration file and to all files within the directory:

```
sudo chown -R kuma:kuma <path>
```

where:

<path> is the path to the directory with the KUMA configuration file.

5. Install the KUMA agent:

```
sudo /opt/kaspersky/kuma/agent agent --core https://<FQDN>:<port> --pair.fingerprint <CA_cert_ID> --id
<service_ID> --wd <path> [--accept-eula]
```

where:

- <FQDN> is the fully qualified domain name of the KUMA Core server.
- <port> is the port that KUMA Core uses for internal communications (port 7210 is used by default).
- <CA_cert_ID> is the fingerprint ID of the CA certificate used for authentication when connecting the service. The `--pair.fingerprint` argument is optional.

  You can install the service without specifying the `--pair.fingerprint` option. In this case, KUMA uses an `insecure` value by default, which skips the validation of the KUMA Core CA certificate when establishing a connection. If the value is `insecure`, the fingerprint does not need to be renewed after reissuing the the CA certificate.

  If you want to specify the fingerprint ID for validation of KUMA Core's CA certificate, go to the Resources and services → Active services section, and in the menu (⋮), select **Copy certificate fingerprint**, then use the copied fingerprint ID in the agent installation command. When reissuing a CA certificate, you need to copy the new certificate fingerprint from the web interface and restart the service with the updated fingerprint ID.
- <service_ID> is the service ID copied from the KUMA web interface.
- <path> is the path to the directory for the files of the KUMA agent that is being installed. If this option is not specified, the files are stored in the directory where the kuma file is located.

You can install two KUMA agents on the same Linux device. In this case, KUMA agents will work in parallel. When installing the second KUMA agent, you need to specify a separate directory for it using the `--wd` option.

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 KUMA 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 KUMA agent.

Examples of installing the KUMA agent:

- Installing the KUMA agent without automatically accepting the End User License Agreement:

```
sudo /opt/kaspersky/kuma/agent agent --core https://kuma.example.com:7210 --id XXXX --wd
/opt/kaspersky/kuma/agent/XXXX
```

- Installing the KUMA agent with automatic acceptance of the End User License Agreement:

```
sudo /opt/kaspersky/kuma/agent agent --core https://kuma.example.com:7210 --id XXXX --wd
/opt/kaspersky/kuma/agent/XXXX --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.

6. If you chose KUMA 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:

```
./agent license --show
```

The KUMA agent is installed on the Linux device.

You can configure the collector to receive data that the KUMA agent sends to KUMA.

Installing a KUMA agent on Windows assets

Before installing the KUMA agent on a Windows device, make sure that the user account under which the KUMA agent will run has the necessary rights.

Rights for a user account when installing a KUMA agent of any type

The server administrator must create a user account with the 'Log on as a service' permission on the Windows asset. This user account must be used to install the agent.

Rights for a user account when installing a KUMA agent of the wec type

The server administrator must create a user account with the 'Event Log Readers' and 'Log on as a service' permissions on the Windows asset. This user account must be used to install and start the agent.

If you plan to receive events from the Security log on a domain controller, grant access to the Security log to the preset Network Service account in Windows. To do this, run the following command on the PowerShell command line of the domain controller device:

```
wevtutil sl security /ca:'O:BAG:SYD:(A;;0xf0005;;;SY)(A;;0x5;;;BA)(A;;0x1;;;S-1-5-32-573)
(A;;0x1;;;S-1-5-20)'
```

If you do not configure access to the Security log, but add the Security log in the wec connector settings, then the stream of events from the KUMA agent will not be received.

If you want to receive events from something other than a domain controller, it is sufficient to add the preset Network Service user account in Windows to the Event Log Readers group.

Rights for a user account when installing a KUMA agent of the wmi type

The server administrator must create a user account with the 'Event Log Readers' and 'Log on as a service' permissions on the Windows asset. This user account must be used to install the agent.

The 'Event Log Readers' and 'Log on as a service' permissions are required to run the agent under a domain account.

If you want to run the agent under a local account, you will need administrator rights as well as 'Event Log Readers' and 'Log on as a service' permissions.

If you want to perform remote collection of events (collection of events from devices connected to the device on which the KUMA agent is installed) and only read logs under a domain account, the 'Event Log Readers' permission is sufficient.

Rights for a user account when installing a KUMA agent of the etw type

The server administrator must create a user account with the 'Log on as a service' permission on the Windows asset. This user account must be used to install the agent.

To run the agent, the user account must have administrator rights or must be added to one of the following security groups:

- Domain admins
- Performance Log Users

Installing the KUMA agent

To install a KUMA agent to a Windows asset:

1. Copy the kuma.exe file to a folder on the Windows asset. `C:\Users\<User name>\Desktop\KUMA` folder is recommended for installation.

   The kuma.exe file is located inside the installer in the /kuma-ansible-installer/roles/kuma/files/ directory.

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 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 [--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 KUMA agent.

   Examples:

   - Installing the KUMA 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 KUMA 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.

     In the command, specify the file containing the password as follows: `< name of the password file >.txt`. For example, `password.txt`.

     ```
     kuma agent --core https://kuma.example.com:7210 --id XXXXX --user domain\username --install < password.txt
     ```

     All data from the password.txt file 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\KUMA\agent\${agent_id}.

   If you did not accept the End User License Agreement (EULA) before installing the agent with the `< password.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 < 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 an `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 < 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\KUMA\agent\<agent ID>` folder is created and the KUMA agent service is installed in it. The agent forwards Windows events to KUMA, 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 KUMA web interface, but only when the service is active. Otherwise, the service needs to be manually restarted on the Windows asset.

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 agent --core https://<fully 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>
```

Update agents

When updating KUMA versions, the WMI, WEC, and ETW 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 web interface, 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

6. When when installing the updated agent on a device for the first time, you must confirm the license. During the installation process, you are prompted to read the text of the license and then accept or reject the agreement. If this did not happen automatically, you can use the following command to view the text of the license:

   ```
   kuma.exe license --show
   ```

   If you want to accept the license agreement, run the command and press `y`:

   ```
   kuma.exe license
   ```

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 (SHA-256) 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 resource set for a KUMA agent.

   At this step, you must create a resource set 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 web interface 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:

1. In the KUMA web interface, under **Resources** → **Agents**, select the required 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 of 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.

   - The /opt/kaspersky/kuma/kuma executable file (the `kuma` file can be found inside the installer in the /kuma-ansible-installer/roles/kuma/files/ directory).

2. Execute the following command:

   ```
   sudo ./agent agent --cfg <config_path> --wd <directory_path> --install
   ```

   where:

   - <config_path> is the path to the agent configuration file.
   - <directory_path> is the path to the directory for the files of the agent that is being installed. If this option is not specified, the files are 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.

   We recommend using 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 computers to KUMA

To transfer events from Windows computers 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 computer 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.
   - Using the ETW connector: the agent connect to the DNS server using the session name and provider specified in the connector settings, 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.

The ETW agent is used only to retrieve events from Windows logs of DNS servers.

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, WMI, or ETW 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-jetstream, 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. **Create an agent service in the KUMA web interface.**

3. **Installing the KUMA agent on the Windows computer 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.

Importing a file with a list of servers for WMI agents

If you have many servers, you can import a .TSV file with a list of event sources for WMI agents so that you do not have to manually fill in the **Server**, **Domain**, and **Log type** settings for the WMI agent for each server. In such a file, you can prepare a list of servers to which the WMI agent will connect and a list of logs that you want to receive from the specified servers.

Requirements for the file with a list of servers for WMI agents.

Acceptable file format: TSV.

If you change the recommended format to **All files** and upload a file that does not comply with the required format (for example, a JSON file, XML file, or some different format instead of TSV), such a file may be parsed incorrectly.

The maximum number of lines in a file with host settings is 500.

The following fields are required:

- **Server** is the name of the host from which the agent will receive logs. Acceptable format: hostname or IP address. Acceptable values: IPv4, IPv6, host name.
- **Domain** is the host ID, for communication with the host over the network. Acceptable format: fully qualified domain name or IP address. Acceptable values: IPv4, IPv6, FQDN.
- **Log type** is a list of logs and/or Windows subscriptions that the agent will receive. Acceptable format: array of strings. For example, you can specify Application, ForwardedEvents, or Security. If the file specifies a value that was not previously present in the list of logs, the log or subscription must be created that will also appear in the list of available for processing using the Windows agent.

  The maximum length of a log value is 1024 characters. You must specify at least one log or subscription.

After importing and filling in all the fields of the remote hosts, a check is performed when you save settings. If any errors are found in the fields, the settings are not saved, and the errors are highlighted. You can manually update the fields as necessary and save the settings.

To import a file with a list of sources:

1. In the **Resources** → **Agents** section of the KUMA interface, create a WMI agent or open the WMI agent settings for editing.
2. This opens a window; in that window, on the **Configuration resource #** tab, in the **Connector** section, specify the name of the connector, in the **Type** drop-down list, select **wmi** and click the **Import TSV** button.
3. This opens a window; in that window, select the prepared file with the list of sources in the .TSV format.

   KUMA parses the file and fills in under **Remote hosts**, fills the **Server**, **Domain**, **Log type** required fields.
4. If necessary, in the **Secret** drop-down list, select a secret of the **Credentials** type or create a secret of the **Credentials** type.

   When importing a list of hosts from a file, in the **Secret** drop-down list, **Default** remains selected, and the agent uses the default credentials when communicating with the server to retrieve logs.
5. If you want to cancel the import of data from the file, click **Cancel**. If you want to delete some data lines imported from a file, in the user interface, select the lines you want to delete and click **Delete**.

6. Save settings in one of the following ways:

   - For a newly created agent, click **Create**. The service is created in the KUMA web interface.
   - For an active agent opened for editing, click **Save**.

7. Apply the saved settings in one of the following ways:

   - For a newly created agent, install the agent service in the KUMA network infrastructure.
   - For an active agent service opened for editing, restart the agent service.

The file with a list of servers for WMI agents is imported.

AI services

This section provides information about the AI services that KUMA works with.

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: $0 \leq score < 0.25$

Medium: $0.25 \leq score < 0.5$

High: $0.5 \leq score < 0.75$

Critical: $0.75 \leq score \leq 1$

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 and removing 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. Save the KUMA Core certificate file in the roles/mlservice/files directory under the installer directory.

6. Change to the directory with the service files and from that directory, run the following command:

   ```
   ./install < path to inventory.yaml >
   ```

7. 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.

8. 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.

Setting	Description
`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.

Analyze with KIRA

In KUMA, you can use Kaspersky Investigation & Response Assistant (KIRA) to analyze the command that triggered the correlation rule. The command is written to the event field if normalization is configured to write the command to the event field. You can view the command in the event card or the correlation event card and click **Analyze with KIRA** in the upper part of the event card to send a query to KIRA. KIRA performs deobfuscation and displays the cached result of the previous request for the command if such a request was performed earlier. This helps investigate alerts and incidents. The analysis results are kept in cache for 14 days and are available for review. Each time a request is sent, an audit event is generated.

This 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. 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.

- The user has one of the following roles with corresponding access rights: General administrator, Administrator, Tier 2 analyst, Tier 1 analyst, and Junior analyst. Only a user with the General administrator role can configure the integration.

Configuring integration with KIRA

To configure integration with KIRA:

1. Get a license with the AI module and activate the functionality in KUMA.
2. In the KUMA web console, go to the **Settings** → **Integrations** → **AI services** section.
3. This opens a window; in that window, go to the **Kaspersky Investigation & Response Assistant** tab.
4. Enable integration with KIRA by enabling the **Status** toggle switch.

   The toggle switch is turned off by default.
5. Click **Select file** and download the <customer name> .ZIP archive, which contains the certificate in PFX format.
6. In the **Certificate password**, enter the password.
7. In the **Proxy server** drop-down list, select a previously created resource or create a new resource.

8. 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.

9. 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 analysis. Analysis is available for all events: new and previously received.

Analyzing using KIRA

After configuring the integration, you can analyze commands using KIRA.

To perform an analysis:

1. Go to the card of the event or correlation event and on the toolbar in the event card, in the **Analyze with KIRA** drop-down list, select the field whose value you want to analyze.

2. This opens the **Analyze with KIRA** window, displaying the command to be analyzed. You can do the following:

   - If the command is obfuscated, it is de-obfuscated automatically without spending tokens. If you want to analyze the command in obfuscated form, in the **Actions** drop-down list, select **Revert to original string**. If necessary, you can de-obfuscate the string again.

   - If you want to know in advance how many tokens will be spent on analysis, in the **Actions** drop-down list, select **Calculate size in tokens**. Number of tokens for analysis = number of tokens to send a request + number of tokens to produce a response.

   - To analyze the command, click the **Analyze** button.

     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 command is analyzed.

The result of the analysis is available in the same **Analyze with KIRA** window: the output, a brief summary, and a detailed analysis. You can also view the result in a separate window by clicking **View result** in the pop-up notification. This opens a separate **KIRA result** window, from which you can also click the link to **Go to event**. After the analysis is completed, the **Result** is displayed on the **KIRA analysis** tab in the event card and is available for viewing by all users with access to the **Analyze with KIRA** functionality.

You can also view the result of the analysis in the **Task manager** section in the properties of the **Query to KIRA** task. You can click the name of the task to select one of the following commands in the context menu:

- **View result** shows the results of the task from the cache to any user with access to KIRA tasks; no tokens are expended.
- **Restart** performs the analysis disregarding the data of the previous analysis stored in the cache; the analysis expends tokens.

Possible errors of the Analyze using KIRA task

Possible errors

HTTP code	Description
400	Invalid client certificate.
404	Error in request.
401	Missing certificate information. Please contact Technical Support.
403	Daily limit of tokens exhausted.
413	Maximum number of tokens for request reached. Make the request smaller.
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, the AI module is used. The AI module analyzes the launch and runtime parameters of applications and identifies suspicious launches of legitimate software with malicious libraries.

To get a status, 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 status, 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 status is received.
- `2` – **Suspicious**. Getting such a result creates an alert if you have the corresponding correlation rule configured.
- `3` – **Bad**. Getting such a result creates an alert if you have the corresponding correlation rule configured. The **Bad** status represents a higher-likelihood detection of DLL Hijacking than the **Suspicious** status.

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 building this type of enrichment into the correlator rather than into the collector. In the correlator, the correlation event is checked, and in the collector, the base event. There are much fewer correlation events than base events, so much fewer queries are needed to check correlation events. 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 Statement in the **Kaspersky Security Network** section (**Settings** → **Integrations** → **AI services** → **DLL Hijacking detection**). If your instance of KUMA has multiple General administrators, it is sufficient for one of them to accept the KSN Statement.

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 solution 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. We recommend using the [OOTB] KEDR Telemetry normalizer imported from the repository.
4. In the **Correlation** section, create a simple or standard correlation rule. In the correlation rule, specify the same fields that you specified when adding the enrichment. 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.
5. 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.
6. If you want to configure the solution to generate an alert when it gets a **Suspicious** or **Bad** 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`

DLL Hijacking attack detection is configured. If you have created a correlation rule, an alert is generated when a **Suspicious** or **Bad** verdict is received.

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 web interface of the solution, 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 solution to generate an alert when it gets a **Suspicious** or **Bad** 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`

DLL Hijacking attack detection is configured. If you have created a correlation rule, an alert is generated when a **Suspicious** or **Bad** status is received.

Metrics service

To monitor the performance of KUMA services and the flow of events from event sources, KUMA collects and stores a large amount of data. The Metrics service is used for collection, storage and analysis of data.

Metrics are available on KUMA services at all times. The Metrics service collects the values of these metrics with an interval of 15 seconds. If the KUMA service is unavailable, changes that are multiples of `<duration of unavailability>` / `<collection interval>` will be lost.

Starting with KUMA 4.2, during the initial installation of KUMA, the Metrics service is installed on the host specified in the `kuma_metrics` section of the inventory file. You can also install the Metrics service on a host of your choice. In any KUMA configuration, only one Metrics service must exist. When upgrading from KUMA 4.0 to 4.2, you must specify the host on which the active Metrics service is running in the `kuma_metrics` section of the inventory file. When the update starts, a check is performed, and if the Metrics service is not found on the specified host, an error is returned and the update is stopped.

In a high-availability configuration of KUMA, the Metrics service communicates with the available KUMA Core directly, bypassing the balancer.

If a Metrics service had been used, and after its removal a new service was installed, historical data is not automatically deleted. The data of the Metrics service remains in the /opt/kaspersky/kuma/metrics/<ID of the old Metrics service instance>/data directory. To migrate the data of the Metrics service to a different host, copy it before the first start of the new service, retaining the access permissions. If necessary, replace the existing data directory with the migrated directory (you need to stop the service before doing this).

Installing the Metrics service on a host

You can install the Metrics service using the expand.sh installer and the expand.inventory.yml inventory file.

To install the Metrics service on a host:

1. Ensure that the target machines meet hardware, software, and installation requirements.
2. On the control computer, go to the directory with the extracted KUMA installer by running the following command:

   ```
   cd kuma-ansible-installer
   ```

3. Create an inventory file named expand.inventory.yml by copying the expand.inventory.yml.template file:

   ```
   cp expand.inventory.yml.template expand.inventory.yml
   ```

4. Edit the expand.inventory.yml inventory file and specify the host on which you want to install the Metrics service in the `kuma_metrics` section.
5. On the control computer, run the following command as root from the directory with the unpacked installer:

   ./expand.sh expand.inventory.yml

   This command creates files for creating and installing the Metrics service on the host specified in the `kuma_metrics` section.
6. In the KUMA web interface, in the **Resources and services** → **Active services** section, in the **Add** drop-down list, select **Add metrics**.
7. This opens the **Add metrics service** window; in that window, specify the name of the service in the **Metrics name** field.
8. Click **Add**.

   The Add metrics service window is minimized, the "Metrics service added" notification is displayed in the upper right corner, and you are taken to the **Active services** section. The Metrics service has been added to the list of services and has the red status.
9. In the Active services section, select the check box next to the Metrics service and in the toolbar or the menu (⋮) select **Copy ID**.
10. Install the Metrics service on the host specified in the expand.inventory.yml inventory file. To do this, run the following command in the console:

    ```
    sudo /opt/kaspersky/kuma/kuma metrics --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 web interface> --api.port <port used for communication with the installed component> --install
    ```

    ```
    Example: sudo /opt/kaspersky/kuma/kuma metrics --core https://kuma.example.com:7210 --id XXXX --api.port YYYY --install
    ```

11. If you want to migrate the historical data of the old Metrics service, copy the 'data' directory from the old host to the new one, preserving all file attributes (owner, group, access permissions, timestamps):

    ```
    sudo mv /opt/kaspersky/kuma/metrics/<ID of the old Metrics service>/data
    /opt/kaspersky/kuma/metrics/<ID of the new Metrics service>/data
    ```

    After copying, make sure that the 'data' directory and the files it contains are owned by the user and group of the Metrics service (by default, kuma:kuma):

    sudo chown -R kuma:kuma /opt/kaspersky/kuma/metrics/<ID of the new Metrics service>/data

    sudo chmod -R 750 /opt/kaspersky/kuma/metrics/<ID of the new Metrics service>/data

12. Start the Metrics service using the following command:

    ```
    sudo systemctl start metrics.service
    ```

The Metrics service is installed and running.

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. **Configuring the KUMA collector for receiving Auditd events**
2. **Configuring the event source server**

   You can configure the event source server using the rsyslog or syslog-ng service.
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 web interface.

Configuring the KUMA collector for receiving Auditd events

At the **Transport** step, select the **TCP** or **UDP** connector type and move the **Auditd** toggle switch to the enabled position.

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 using the rsyslog service

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:

   ```
   yum install rsyslog
   ```

   ```
   systemctl enable rsyslog.service
   ```

   ```
   systemctl start rsyslog.service
   ```

2. Open the audit.service configuration file, /etc/audit/auditd.conf, and change the values of the following settings:

   ```
   name_format=NONE
   ```

   ```
   log_format=ENRICHED
   ```

   The `log_format` parameter determines the format of the records generated by the auditd service, and must be set to `ENRICHED` to have user information saved in events.

   After editing the settings, restart the auditd service:

   ```
   sudo systemctl restart auditd.service
   ```

3. In the /etc/rsyslog.d directory, create the audit.conf file with the following content, depending on your protocol:

   - To send events over TCP:

     ```
     $ModLoad imfile
     $InputFileName /var/log/audit/audit.log
     $InputFileTag tag_audit_log:
     $InputFileStateFile audit_log
     $InputFileSeverity info
     $InputFileFacility local6
     $InputRunFileMonitor
     *.* @@<IP address of the KUMA collector>:<port of the KUMA collector>
     For example:
     *.* @@192.1.3.4:5858
     ```

   - To send events over UDP:

     ```
     $ModLoad imfile
     $InputFileName /var/log/audit/audit.log
     $InputFileTag tag_audit_log:
     $InputFileStateFile audit_log
     $InputFileSeverity info
     ```

```
$InputFileFacility local6
$InputRunFileMonitor
template(name="AuditFormat" type="string" string="<%PRI%>%TIMESTAMP:::date-rfc3339%
%HOSTNAME% %syslogtag% %msg%\n")
*.* @<IP address of the KUMA collector>:<port of the KUMA collector>
For example:
*.* @192.1.3.4:5858;AuditFormat
```

4. Save the changes to the audit.conf file.
5. 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 the event source server using the syslog-ng service

The syslog-ng 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 syslog-ng service is installed on the event source server by running the following command:

   ```
   systemctl status syslog-ng.service
   ```

   If the syslog-ng service is not installed on the server, install the service by running one of the following commands depending on your package manager:

   - For APT (Debian, Ubuntu):

     ```
     apt install syslog-ng
     ```

   - For YUM (CentOS 7, RHEL):

     ```
     yum install syslog-ng
     ```

   - For DNF (Fedora, CentOS 8):

     ```
     dnf install syslog-ng
     ```

   - For Zypper (openSUSE):

     ```
     zypper install syslog-ng
     ```

- For Pacman (Arch Linux):

```
pacman -S syslog-ng
```

- For Snap (if available):

```
snap install syslog-ng
```

2. After installation, start the syslog-ng service:

```
systemctl enable syslog-ng.service
systemctl start syslog-ng.service
```

3. Open the /etc/audit/auditd.conf file of the audit.service service and change the value of the `name_format` setting to `NONE`:

   `name_format=NONE`

4. After editing the settings, restart the auditd service:

```
sudo systemctl restart auditd.service
```

5. In the /etc/syslog-ng directory, create the audit.conf file with the following content, depending on your protocol:

   - To send events over TCP:

```
source s_audit_log {
  file("/var/log/audit/audit.log"
    persist-name("s_audit_log_for_siem")
    follow-freq(1)
    program-override("auditd:")
    flags(no-parse));
};
log {
  source(s_audit_log);
  destination(d_audit_log);
};
```

```
# Message formatting
template AuditFormat {
  template("<${PRI}>${ISODATE} ${HOST} ${MSG}\n");
};
```

```
# Transferring audit logs to the specified KUMA collector address
destination d_audit_log {
  tcp("<KUMA collector IP address>" port(<KUMA collector port>) persist-
  name("collector-audit") template(AuditFormat));
```

```
};
```

- To send events over UDP:

```
source s_audit_log {
  file("/var/log/audit/audit.log"
    persist-name("s_audit_log_for_siem")
    follow-freq(1)
    program-override("auditd:")
    flags(no-parse));
};
log {
  source(s_audit_log);
  destination(d_audit_log);
};
```

```
# Message formatting
```

```
template AuditFormat {
  template("<${PRI}>${ISODATE} ${HOST} ${MSG}\n");
};
```

```
# Transferring audit logs to the specified KUMA collector address
```

```
destination d_audit_log {
  udp("<KUMA collector IP address>" port(<KUMA collector port>) persist-
  name("collector-audit") template(AuditFormat));
};
```

6. Save the changes to the audit.conf file.

7. Add the following line to /etc/syslog-ng/syslog-ng.conf:

```
@include " audit.conf "
```

8. Restart the syslog-ng service by executing the following command:

```
systemctl restart syslog-ng.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 web interface, make sure that the port number and connector type match the port and protocol specified in the Kaspersky Anti Targeted Attack Platform event export settings in KUMA.

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. Creating a KUMA collector
3. Installing the KUMA collector in the network infrastructure
4. 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 web interface. 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 web interface.

For details on creating a KUMA collector, refer to Creating a collector.

When creating a collector in the KUMA web interface, 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 the export of Kaspersky Security Center events to the KUMA SIEM system

KUMA allows you to receive and export events from the Kaspersky Security Center Administration Server to the KUMA SIEM system.

Configuring the export and receipt of Kaspersky Security Center events involves of the following steps:

1. Configuring the export of Kaspersky Security Center events.
2. Configuring the KUMA Collector.
3. Installing the KUMA collector in the network infrastructure.
4. Verifying the receipt of Kaspersky Security Center events by KUMA.

   You can verify if the events from Kaspersky Security Center Administration Server were correctly exported to the KUMA SIEM system by using the KUMA web interface to search for related events.

To display Kaspersky Security Center events in the table, enter the following search expression:

```
SELECT * FROM `events` WHERE DeviceProduct = 'KSC' ORDER BY Timestamp DESC LIMIT 250
```

Configuring export of Kaspersky Security Center events in CEF format

Kaspersky Security Center allows you to configure the settings for exporting events in the CEF format to a SIEM system.

The function of exporting Kaspersky Security Center events in the CEF format to SIEM systems is available with Kaspersky Endpoint Security for Business Advanced license or above.

To configure export of events from Kaspersky Security Center Administration Server to the KUMA SIEM system:

1. In Kaspersky Security Center console tree, select the **Administration server** node.
2. In the workspace of the node, select the **Events** tab.
3. Click the **Configure notifications and event export** link and select **Configure export to SIEM system** from the drop-down list.

   The **Properties: Events** window opens. By default the **Events export** section is displayed.
4. In the **Events export** section, select the **Automatically export events to SIEM system database** check box.
5. In the **SIEM system** drop-down list select **ArcSight (CEF format)**.
6. In the corresponding fields, specify the address of the KUMA SIEM system server and the port for connecting to the server. Select **TCP/IP** as the protocol.

   You can click **Export archive** and specify the starting date from which pre-existing KUMA events are to be exported to the SIEM system database. By default, Kaspersky Security Center exports events starting from the current date.
7. Click **OK**.

As a result, the Kaspersky Security Center Administration Server automatically exports all events to the KUMA SIEM system.

Configuring KUMA collector for collecting Kaspersky Security Center events

After configuring the export of events in the CEF format from Kaspersky Security Center Administration Server, configure the collector in the KUMA web interface.

To configure the KUMA Collector for Kaspersky Security Center events:

1. In the KUMA web interface, 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 Kaspersky Security Center 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 Kaspersky Security Center events

After configuring the collector for collecting Kaspersky Security Center 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 Kaspersky Security Center event from MS SQL

KUMA allows you to receive information about Kaspersky Security Center events from an MS SQL database.

Before configuring, make sure that you have created the KUMA collector for Kaspersky Security Center 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 Kaspersky Security Center 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 web interface. This resource is used to store credentials for connecting to MS SQL.

To create a KUMA secret:

1. In the KUMA web interface, open **Resources** → **Secrets**.

   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 following 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 web interface, select **Resources** → **Connectors**.
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 web interface 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 web interface, 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.



https://www.youtube.com/watch?v=bfCCmmOVwO8

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 **ForwardedEvents** 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** → **ForwardedEvents**.

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. This opens a window; in that 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 web interface 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. This opens the **Basic event parsing** window; in that window, in the **Normalizer** field, select **[OOTB] Microsoft Products** 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 Windows device events, in the KUMA Collector Setup Wizard, at the **Event parsing** step, in the **Normalizer** field, select **[OOTB] Microsoft Products**.
6. Installing the 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.



https://www.youtube.com/watch?v=bfCCmmOVwO8

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. This opens a window; in that 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 DNS server events using the ETW agent

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:

- *Providers* are elements of the system that generate events and send them 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).

- *Consumers* are software systems that receive events generated by providers from the ETW connector, and use these events in some way. For example, KUMA can act as a consumer.

- *Controllers* are pieces of software that manage 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 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 changes to green in the KUMA web interface.

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 so, right-click next to the KUMA agent service and 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 (**Log on as a service**).

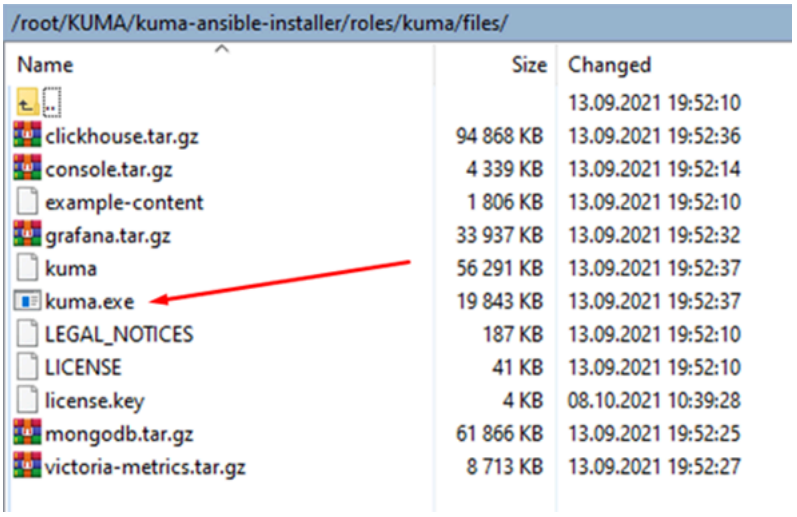
7. Installing a KUMA agent on a Windows server.

   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 web interface, 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 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 web interface, 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 web interface.

Configuration on the Windows side

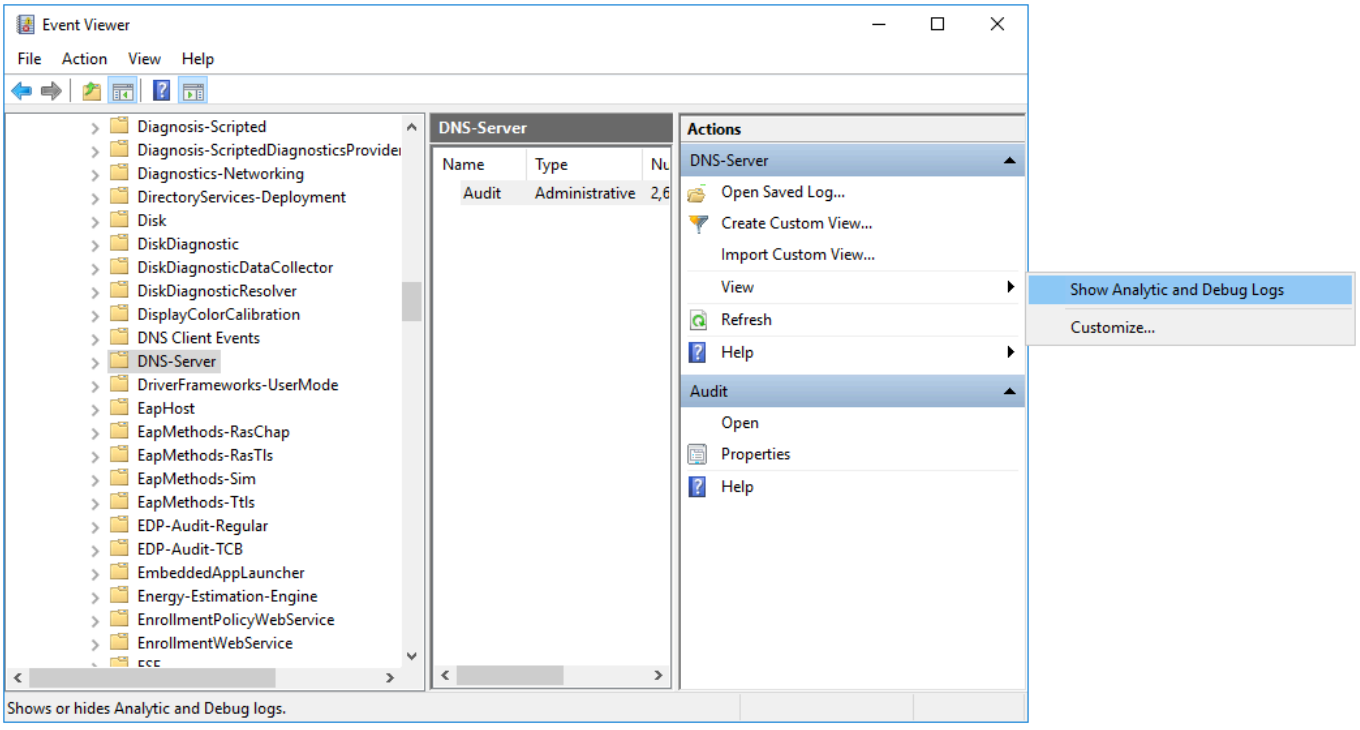
To configure the reception of DNS server events using the ETW connector on the Windows side:

1. Start the Event viewer by running the following command:

   ```
   eventvwr.msc
   ```

2. This opens a window; in that window, 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**.

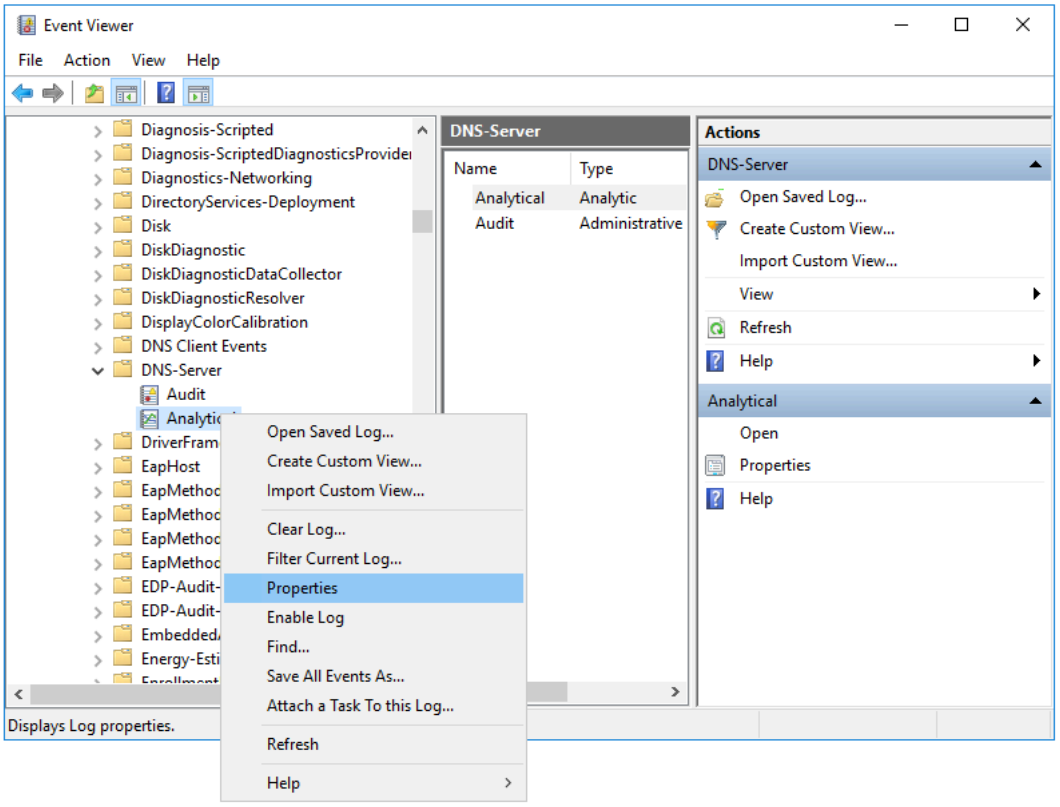


Displaying the analytical and debug logs

The **Audit** debug log and **Analytical** log are displayed.

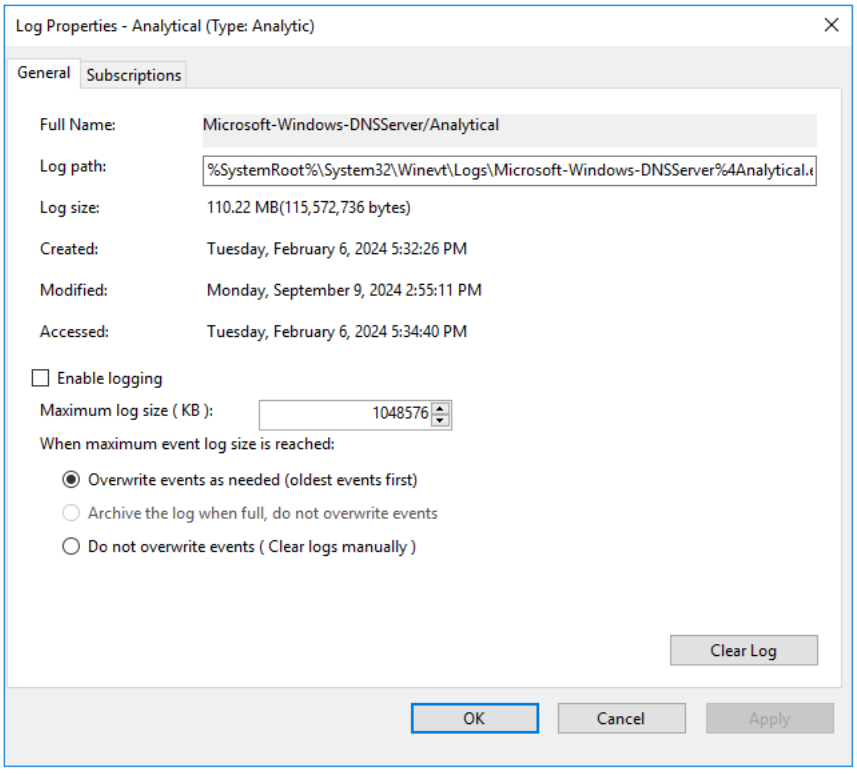
4. Configure the analytic log:

a. Open the context menu of the **Analytical** log and select **Properties**.



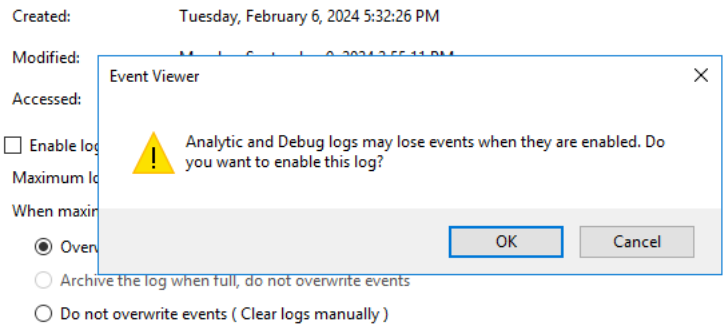
Go to the properties of the Analytical log

a. This opens a window; in that window, make sure that in the **Max Log Size (KB)** field, the value is `1048576`.



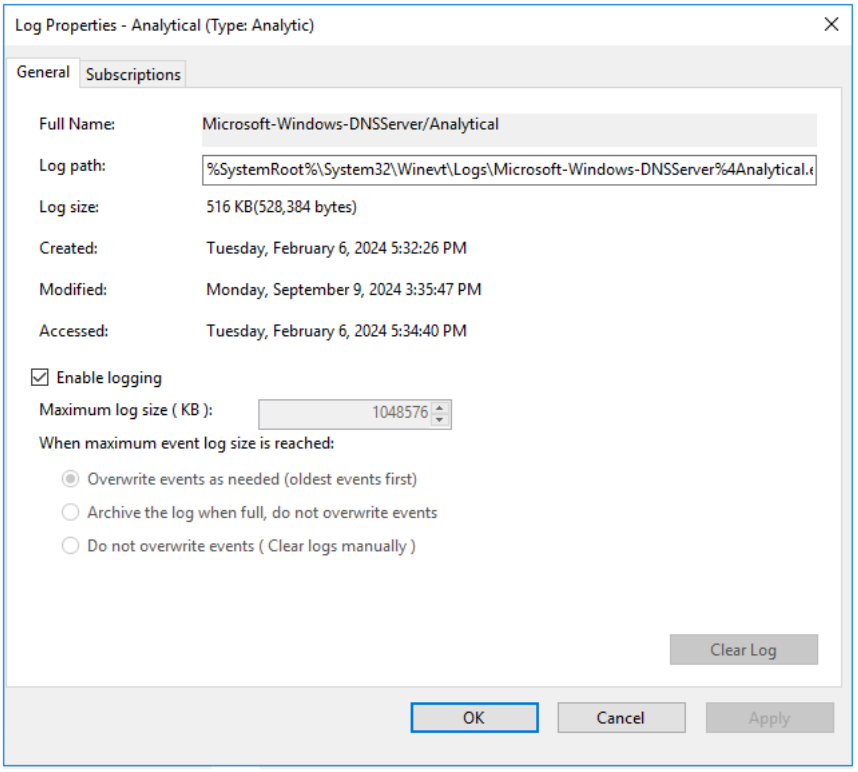
Analytical log properties

a. Select the **Enable logging** check box and in the confirmation window, click **OK**.



Confirmation window

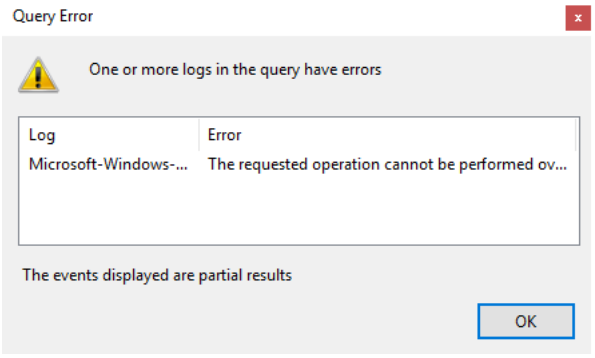
The analytic log must be configured as follows:



Reviewing the settings of the Analytical log for correctness

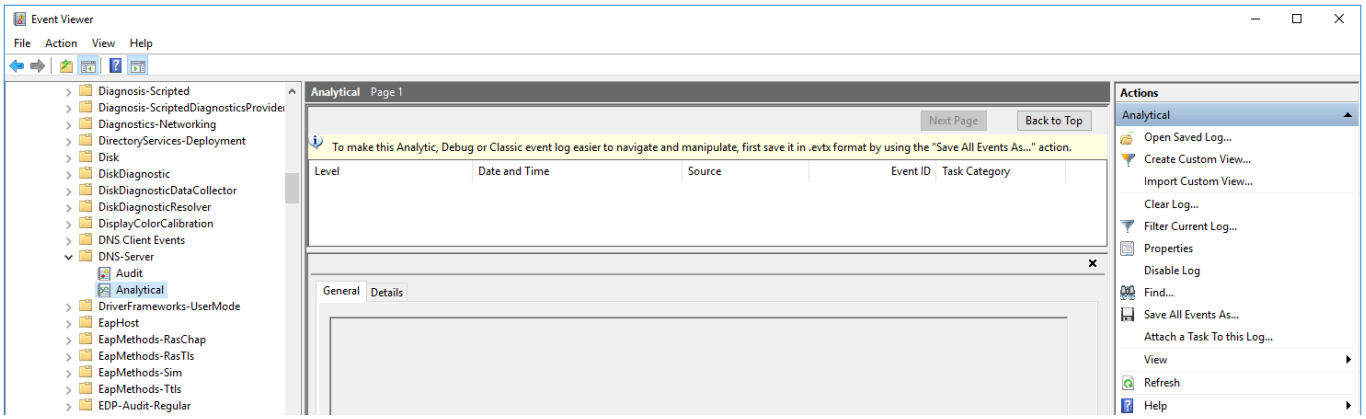
a. Click **Apply**, then click **OK**.

An error window is displayed.



Error window

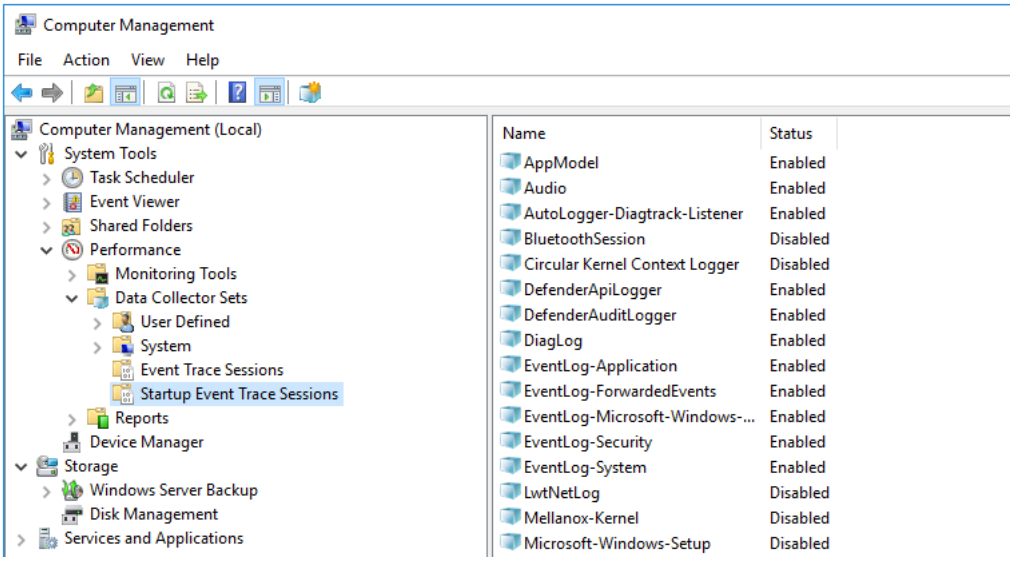
When analytic log rotation is enabled, events are not displayed. To view events, in the **Actions** pane, click **Stop logging**.



Stopping logging

5. Start Computer management as administrator.

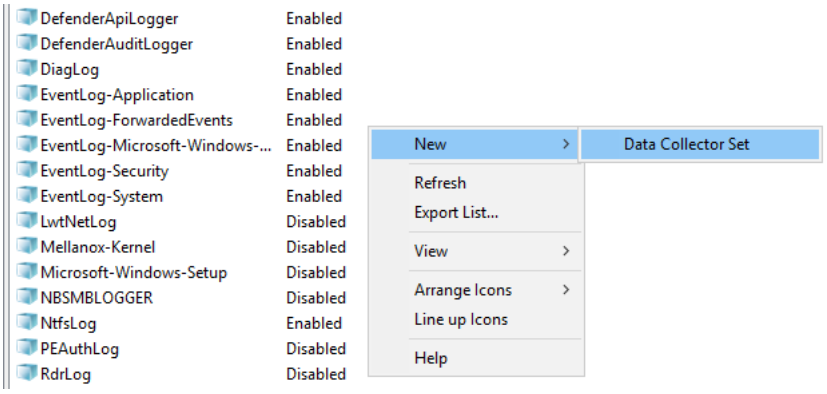
6. This opens a window; in that window, go to the **System Tools** → **Performance** → **Startup Event Trace Sessions** folder.



Managing the computer

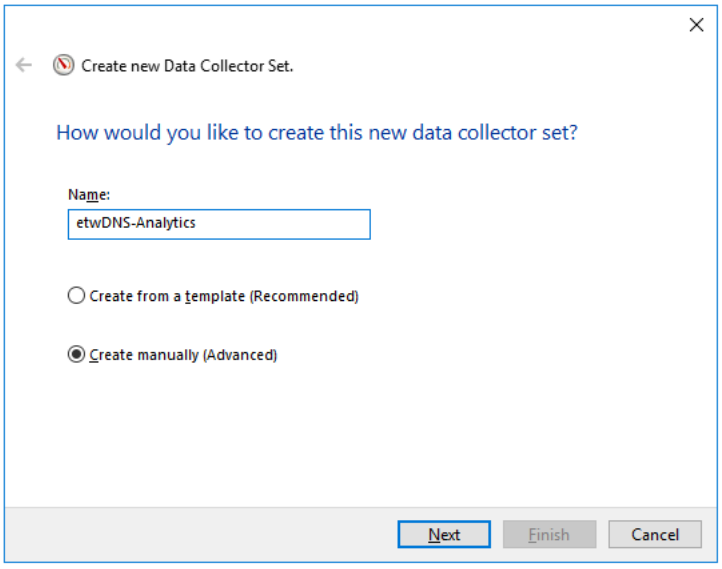
7. Create a provider:

a. Open the context menu of the **Startup Event Trace Sessions** folder and select **Create → Data Collector Set**.



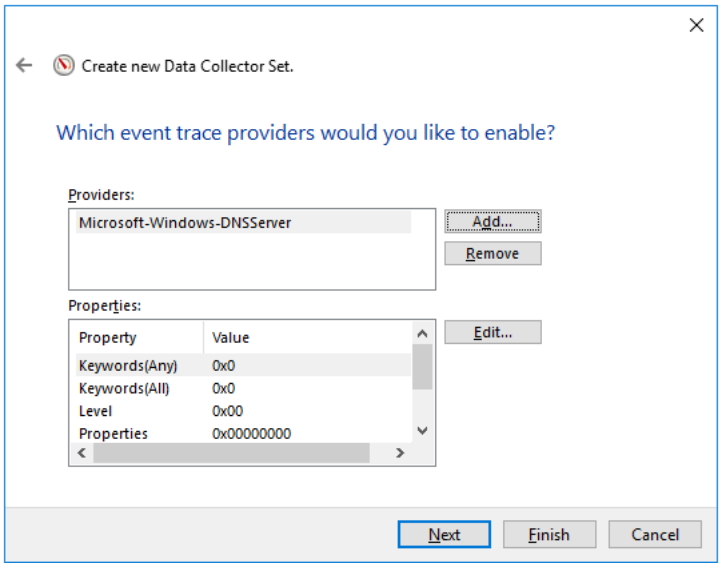
Creating a data collector group

a. This opens a window; in that window, enter the name of the provider and click **Next**.



Entering the provider name

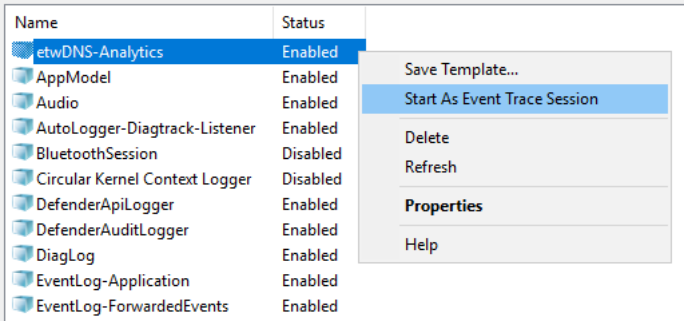
a. Click **Add...** and in the displayed window, select the **Microsoft-Windows-DNSServer** provider.



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**.



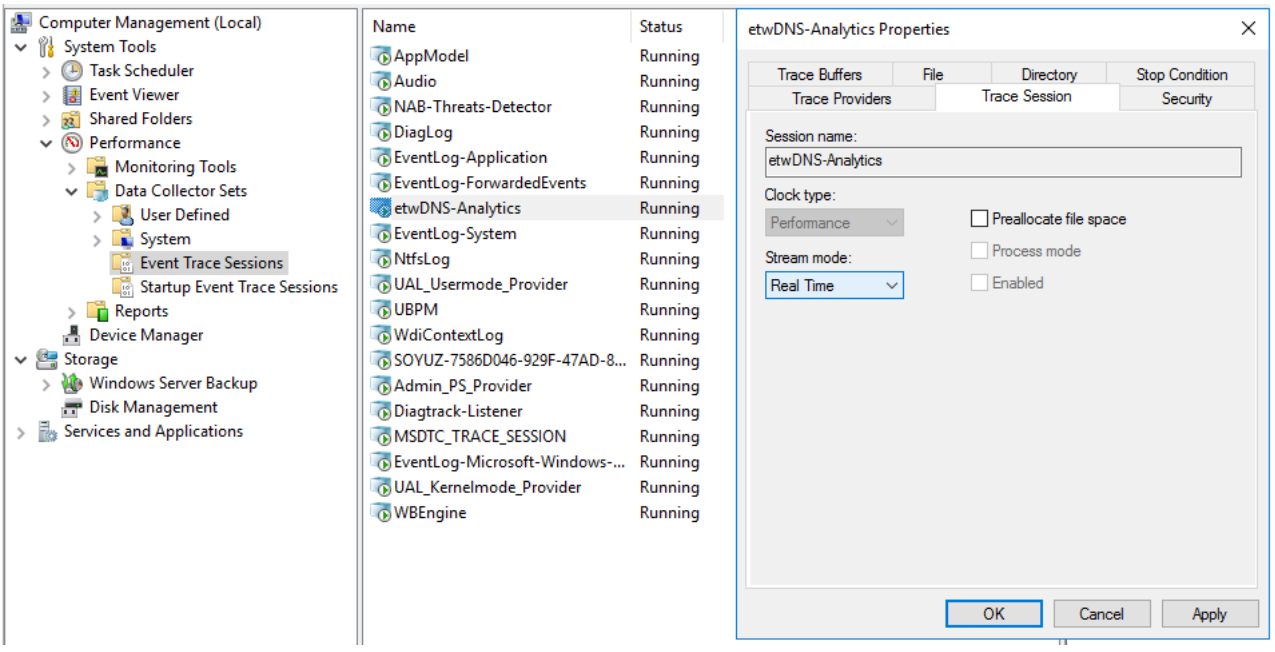
Starting the provider

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. This opens a window; in that window, select the **Trace Sessions** tab and in the **Stream Mode** drop-down list, select **Real Time**.



Configuring stream mode

12. Click **Apply**, then click **OK**.

DNS server event reception using the ETW connector is configured.

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 web interface.

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 PostgresSQL 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 commands:

   ```
   yum install rsyslog
   ```

   ```
   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 @@< IP address of the collector >:< port of the collector >
   ```

   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 web interface.

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 web interface.

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 field

      Syslog Servers
      of the configuration you are configuring, click

      Assign
      .

   d. Select the check boxes for syslog servers that you want to assign and click

      the ✔ icon.

      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 web interface.

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 with administrator rights.
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 web interface.

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. Depending on the version of KLMS you are using, select one of the following options:

   - Configuring export of KLMS 8 events to KUMA
   - Configuring the export of KLMS 10 events to KUMA (documentation for KLMS 10 is available in Russian only).

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 web interface.

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) 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 web interface.

Configuring the receipt of KICS for Networks events

You can configure the receipt of events from Kaspersky Industrial CyberSecurity for Networks (KICS for Networks) 4.2 in KUMA.

Configuring event receiving consists of the following steps:

1. Creating a KICS for Networks connector for sending events to KUMA.
2. Configuring export of KICS for Networks events to KUMA.
3. Creating and installing a KUMA collector to receive KICS for Networks events.
4. Verifying receipt of KICS for Networks events in the KUMA collector.

   You can verify that KICS for Networks event export is correctly configured in the Searching for related events section of the KUMA web interface.

Creating a KICS for Networks connector for sending events to KUMA

To create a connector for sending events in the web interface of KICS for Networks:

1. Log in to the KICS for Networks web interface using an administrator account.
2. Go to the **Settings** → **Connectors** section.
3. Click the **Add connector** button.
4. Specify the following settings:
   a. In the **Connector type** drop-down list, select **SIEM**.
   b. In the **Connector name** field, specify a name for the connector.
   c. In the **Server address** field, enter the IP address of the KICS for Networks Server.
   d. In the **Connector deployment node** drop-down list, select the node on which you are installing the connector.

      You can specify any name.

   e. In the **User name** field, specify the user name for KUMA to use for connecting to the application through the connector. You must specify the name of one of the KICS for Networks users.
   f. In the **SIEM server address** field, enter the IP address of the KUMA collector server.
   g. In the **Port number** field, enter the port number of the KUMA collector.
   h. In the **Transport protocol** drop-down list, select **TCP** or **UDP**.
   i. Select the **Allow sending audit entries** check box.
   j. Select the **Allow sending application entries** check box.
5. Click the **Save** button.

   The connector is created. It is displayed in the table of KICS for Networks connectors with the **Running** status.

The KICS for Networks connector for sending events to KUMA is ready for use.

Configuring export of KICS for Networks events to KUMA

To configure the sending of security events from KICS for Networks to KUMA:

1. Log in to the KICS for Networks web interface using an administrator account.
2. Go to the **Settings** → **Event types** section.
3. Select the check boxes for the types of events that you want to send to KUMA.
4. Click **Select connectors**.
5. This opens a window; in that window, select the connector that you created for sending events to KUMA.
6. Click **OK**.

   Events of selected types will be sent to KUMA. In the **Event types** table, such events are marked with a check box in the column with the connector name.

Creating a KUMA collector to receive KICS for Networks events

After configuring the event export settings, you must create a collector for KICS for Networks events in the KUMA web interface.

For details on creating a KUMA collector, refer to Creating a collector.

When creating a collector in the KUMA web interface, you must:

1. At the **Transport** step, select the transport protocol type matching the type you selected when you created the connector in KICS for Networks at step 4i (TCP or UDP) and the port number matching the port number you specified at step 4h.
2. At the **Event parsing** step, select the **[OOTB] KICS4Net v3.x** normalizer.
3. At the **Routing** step, make sure that the following destinations are added to the collector resource set:

   - **storage**—used to transmit data to the storage.
   - **correlator**—used to transmit data to the correlator.

   If destinations have not been added to the collector, you must create them.
4. At the last step of the wizard, a command is displayed in the lower part of the window, which you can use to install the service on the server that you want to receive events. Copy this command and use it when installing the second part of the collector.

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 web interface.

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 web interface>

   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 an example 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 review the list of current plug-ins.
- `SHOW GLOBAL VARIABLES LIKE 'server_audit%'` to review 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 review the list of current plug-ins.
- `SHOW GLOBAL VARIABLES LIKE 'server_audit%'` to review 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` 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
input(type="imfile"
File="/var/log/httpd/error_log"
Tag="tag_FreeIPA_log_httpd"
facility="local3")
input(type="imfile"
File="/var/log/dirsrv/slapd-*/audit"
Tag="tag_FreeIPA_log_audit"
StartMsg.regex="^time:"
facility="local3")
input(type="imfile"
File="/var/log/dirsrv/slapd-*/errors"
Tag="tag_FreeIPA_log_errors"
facility="local3")
input(type="imfile"
File="/var/log/dirsrv/slapd-*/access"
Tag="tag_FreeIPA_log_access"
facility="local3")
input(type="imfile"
File="/var/log/krb5kdc.log"
Tag="tag_FreeIPA_log_krb5kdc"
facility="local3")
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 KUMA collector >"
port="< port of KUMA collector >"
protocol="< udp or tcp >"
template="ls_json")
stop
}
```

4. Add the following lines to the /etc/rsyslog.conf configuration file:

```
$IncludeConfig /etc/rsyslog.d/freeipa-to-siem.conf
$RepeatedMsgReduction off
```

5. If necessary, disable the local recording of FreeIPA events by doing the following:

   a. Open the rsyslog configuration file /etc/rsyslog.conf or the default rules file /etc/rsyslog.d/50-default.conf for editing.

   b. In that file, find the line that is responsible for recording system events, and add the `local3.none` parameter to it because in the /etc/rsyslog.d/freeipa-to-siem.conf configuration file, the `facility` is set to `local3`. The edited line should look like this:

      ```
      *.info;mail.none;authpriv.none;cron.none;local3.none /var/log/messages
      ```

      If `facility` is set to `local3` for another application on the server, change the `facility` of FreeIPA in the /etc/rsyslog.d/freeipa-to-siem.conf file to a different value and edit the chosen rsyslog file accordingly.

   c. Save your changes.

6. Save changes to the /etc/rsyslog.d/freeipa-to-siem.conf configuration file.

7. 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 web interface.

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, enable the **Do not send incident information in CEF format** toggle switch.

   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 the KUMA SIEM system.

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 web interface.

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 the KUMA SIEM system.

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 web interface.

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 administrator rights.
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 the KUMA SIEM system.

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 web interface.

Configuring audit 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 the KUMA SIEM system.

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 web interface.

Configuring audit of FreeRADIUS events

To configure event audit in the FreeRADIUS system:

1. Connect to the server where the FreeRADIUS system is installed using an account with administrator rights.
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 web interface.

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 with administrator rights.
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 the KUMA SIEM system.

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 web interface.

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 computer 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 the KUMA SIEM system.

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.

   When creating a collector in the KUMA web interface, 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 web interface.

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 using an account with administrator rights.
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 Windows event reception using Kaspersky Endpoint Security for Windows

In Kaspersky Endpoint Security for Windows, starting from version 12.6, events can be sent from Windows logs to a KUMA collector. In this way, KUMA can get events from Windows logs from all hosts with Kaspersky Endpoint Security for Windows 12.6 or later without installing KUMA agents on such hosts.

Kaspersky Endpoint Security for Windows versions 12.6–12.10 supports a limited set of EventIDs from Microsoft products.
In Kaspersky Endpoint Security for Windows versions 12.6–12.12, events with similar parameters are filtered and the first of such events is sent, while subsequent similar events are not sent. If a massive amount of similar events occurs, some of them may be sent; hundreds or thousands of similar events are necessary for this to happen. However, a distinct event, such as a different user logging in, is sent. The filtering logic is defined by Kaspersky experts and allows reducing the volume of transmitted events without losing data that is useful for detecting attacks.

To activate this functionality, you need:

- A valid KUMA license
- Kaspersky Security Center 14.2 or later
- Kaspersky Endpoint Security for Windows 12.6 or later

Configuring event receiving consists of the following steps:

1. **Importing the normalizer into KUMA**

   In KUMA, you must configure getting updates through Kaspersky update servers.

   In the **Resources** section, click **Import resources** and in the list of normalizers available for installation, select **[OOTB] Microsoft Products via KES WIN**.

2. **Creating a KUMA collector for receiving Windows events**

   Depending on the version of Kaspersky Endpoint Security for Windows:

   - If you are using Kaspersky Endpoint Security for Windows version 12.6–12.10, at the **Transport** step, select **TCP** or **UDP** and specify the port number that the connector will listen on. At the **Event parsing** step, select the **[OOTB] Microsoft Products via KES WIN** normalizer, and at the **Event filtering** step, select the **[OOTB] Microsoft Products via KES WIN - Event filter for collector** filter.

   - If you are using Kaspersky Endpoint Security for Windows version 12.11 or later, at the **Transport** step, select **TCP** or **UDP** and specify the port number that the connector will listen on. At the **Event parsing** step, select one of the following normalizers:

     - For KUMA version 3.2–4.0, select the **[OOTB] Microsoft Products via KES WIN NG for KUMA 3.2** normalizer.

     - For KUMA version 4.0.1 or later, select the **[OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1** normalizer.

3. **Requesting a key from Technical Support**

   If your license did not include a key for activating the functionality of sending Windows logs to the KUMA collector, send the following message to Technical Support: "We have purchased a KUMA license and are using Kaspersky Endpoint Security for Windows version 12.6. We want to activate the functionality of sending Windows logs to the KUMA collector. Please provide a key file to activate the functionality." New KUMA users do not need to make a Technical Support request because new users get 2 keys with licenses for KUMA and for activating the Kaspersky Endpoint Security for Windows functionality.

   In response to your message, you will get a key file.

4. **Configuration on the side of Kaspersky Security Center and Kaspersky Endpoint Security for Windows.**

   A key file that activates the functionality of sending Windows events to KUMA collectors must be imported into Kaspersky Security Center and distributed to Kaspersky Endpoint Security endpoints in accordance with the instructions. You must also add KUMA server addresses to the Kaspersky Endpoint Security policy and specify network connection settings.

5. **Verifying receipt of Windows events by the KUMA collector**

   You can verify that the Windows event source server is correctly configured in the Searching for related events section of the KUMA web interface.

The list of events from Microsoft products that Kaspersky Endpoint Security for Windows version 12.6–12.10 sends to KUMA is given in the following table.

Event log	Event identifier
System	12 13 7040 7045 42 104 107 109 1074 6005 6006 7034 7036 8003

Event log	Event identifier
Security	1102 4614 4649 4696 4698 4704 4706 4713 4715 4717 4720 4723 4724 4725 4726 4727 4728 4729 4738 4739 4740 4741 4742 4756 4757 4765 4766 4767 4768 4769 4770 4771 4775 4776 4778 4780 4781 4794 4817 4876 4877 4882 4885 4886 4887 4890 4891 4898 4904 4905 4928 4950 4964 5136 5137 5138 5139 5141 5142 5143 5144 5155 5376 5377 5632 5888 5890 6416 4622 4648 4672 4697 4702 4719 4732 4733

Event log	Event identifier
	4798 4946 4947 4948 4949 5145 4616 4625 4663 4624 4799 5140 1008 1105 2722 4615 4618 4626 4627 4634 4647 4653 4654 4656 4657 4658 4659 4660 4661 4664 4666 4667 4670 4673 4674 4688 4689 4690 4691 4692 4693 4694 4695 4699 4700 4701 4703 4705 4707 4714 4716 4718 4730 4731 4734 4737 4743 4744 4745 4746 4747 4748 4749 4750 4751 4752 4753 4754 4755 4758 4759 4760 4761 4762 4763 4764

Event log	Event identifier
	4772 4773 4774 4777 4779 4782 4783 4784 4785 4786 4787 4788 4789 4790 4791 4792 4793 4797 4800 4801 4802 4803 4818 4819 4820 4821 4822 4823 4824 4825 4826 4865 4866 4867 4868 4869 4870 4871 4872 4873 4874 4875 4883 4884 4888 4892 4893 4896 4906 4907 4908 4911 4912 4913 4929 4930 4931 4932 4933 4935 4944 4945 4951 4953 4956 4957 4958 4981 4982 4983 4984 4985 5024 5031 5033 5039

Event log	Event identifier
	5049 5051 5056 5057 5058 5059 5060 5061 5063 5064 5065 5066 5067 5068 5069 5070 5071 5122 5123 5146 5147 5152 5153 5154 5156 5157 5158 5159 5168 5378 5379 5380 5381 5382 5447 5448 5451 5452 5459 5461 5472 5474 5477 5478 5483 5484 5633 6144 6145 6272 6273 6274 6276 6278 6279 6280 6281 6410 6419 6420 6421 6422 6423 6424 6889

Event log	Event identifier
PowerShell	4100 4103 4104 4105 4106 8193 8194 8197 24577 24595 24596 24597 24598 24599 53249 53250 53504
MS SQL Server	615 919 958 1945 2007 2812 3406 3407 3421 3454 5084 5579 5701 5703 6253 8128 9013 9666 15268 15457 17104 17110 17111 17125 17137 17152 17164 17176 17177 17199 17201 17550 17551 17561 17663 18264 18265 18456 18488 18496 19030 19031 19032 26022 26037 26048 26067 33090 49903 49904

Event log	Event identifier
Microsoft Defender	1006 1015 1116 1117 1000 1001 2000 5000 5001 5002 5004 5007 5010 5012
Terminal Server	1149 21 22 24 25 39 40
Microsoft Active Directory Federation Service (AD FS)	106 217 251 335 342 349 358 364 381 385 400 401 417 424 435 436
Sysmon	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 255
Microsoft Active Directory Domain Service (AD DS)	1213 1317 1644 2041 2889

Configuring Linux event reception using Kaspersky Endpoint Security for Linux

In Kaspersky Endpoint Security for Linux, starting from version 12.2, events can be sent from Linux logs to a KUMA collector. This allows KUMA to receive events from Linux logs from all hosts on which Kaspersky Endpoint Security for Linux version 12.2 is installed. To activate the functionality, you need:

- A valid KUMA license
- Kaspersky Security Center 14.2 or later
- Kaspersky Endpoint Security for Linux 12.2 or later

Configuring the receipt of events involves the following steps:

1. Importing the normalizer into KUMA.

   In KUMA, you must configure getting updates through Kaspersky update servers.

   Click Resource import and select **[OOTB] KESL syslog cef** in the list of available normalizers.

2. Creating a KUMA collector for receiving Linux events.

   To receive Linux 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] KESL syslog cef** normalizer.

3. Requesting a key from Technical Support.

   If your license did not include a key for activating the functionality of sending Linux logs to the KUMA collector, send the following message to Technical Support: "We have purchased a KUMA license and are using Kaspersky Endpoint Security for Linux version 12.2. We want to activate the functionality of sending Linux logs to the KUMA collector. Please provide a key file to activate the functionality." New KUMA users do not need to make a Technical Support request because new users get 2 keys with licenses for KUMA and for activating the Kaspersky Endpoint Security for Linux functionality.

   In response to your message, you will get a key file.

4. Configuration on the side of Kaspersky Security Center and Kaspersky Endpoint Security for Linux.

   A key file that activates the functionality of sending Linux events to KUMA collectors must be imported into Kaspersky Security Center and distributed to Kaspersky Endpoint Security endpoints in accordance with the instructions. You must also add KUMA server addresses to the Kaspersky Security Center policy and specify network connection settings.

5. Verifying receipt of Linux events in the KUMA collector

   You can verify that the Linux event source server is correctly configured in the Searching for related events section of the KUMA web interface.

   Kaspersky Endpoint Security for Linux sends the following events:

   - ProcessCreate
   - ProcessTerminate
   - FileChange
   - New user account created

- User account deleted
- Group created
- Group deleted
- New member added to a group
- Member removed from the group
- User password changed
- Linux authentication performed
- Linux session started
- Linux session finished
- Service started
- Service stopped
- Promiscuous mode modified
- Audit configuration changed
- Account expiration date changed

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 events, 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 web interface.

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 with administrator rights.
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 web interface.

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 resource set 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 to 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.

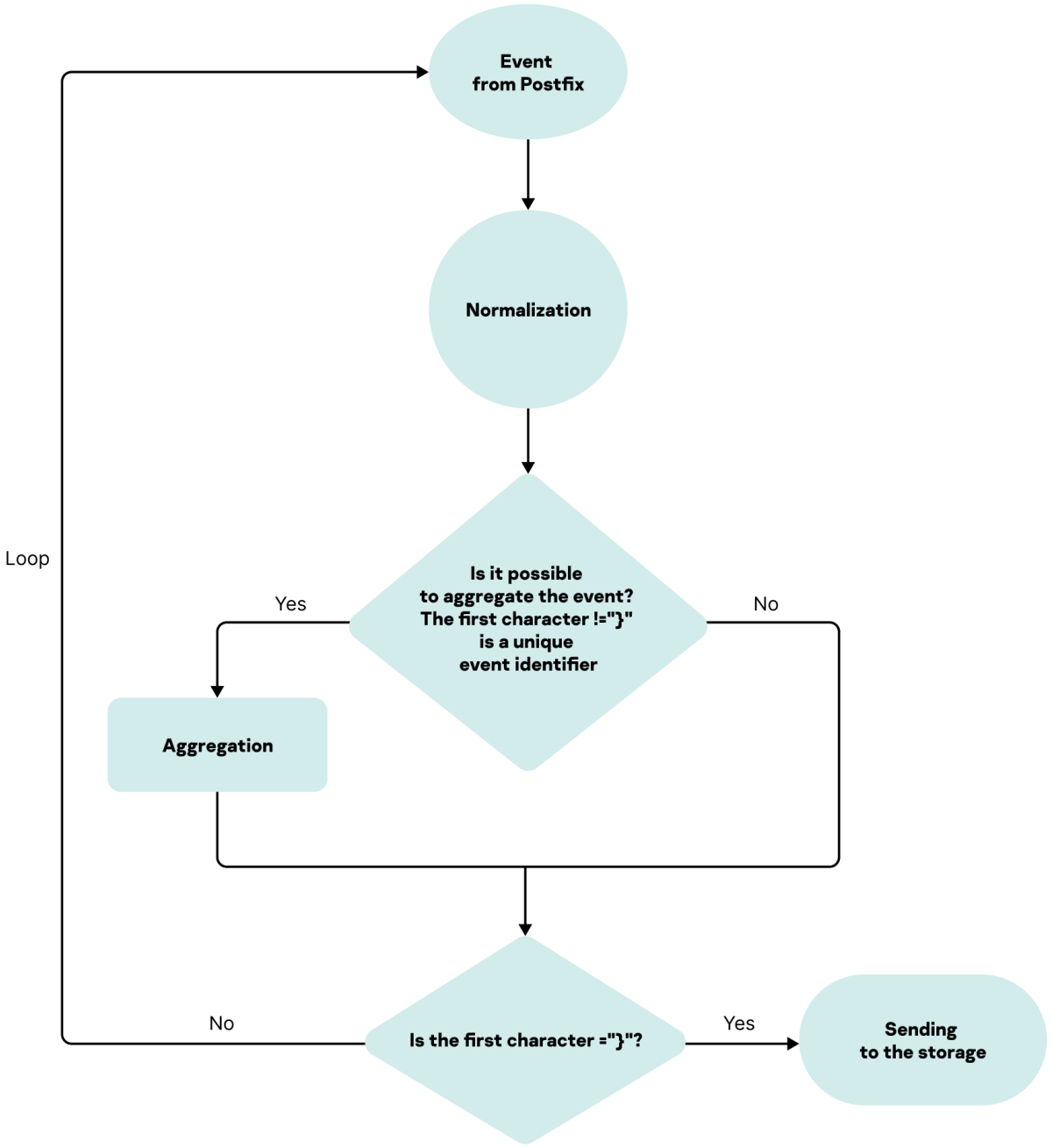
5. Normalization of the aggregated event

   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.



Processing diagram for events from the Postfix system

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:

- 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 **Postfix. Aggregated events to storage and correlator**.

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 **Postfix. Aggregated events to storage and correlator**.

  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 web interface. 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 and 6.5.1 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 some 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 web interface.

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 resource set 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 to 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.

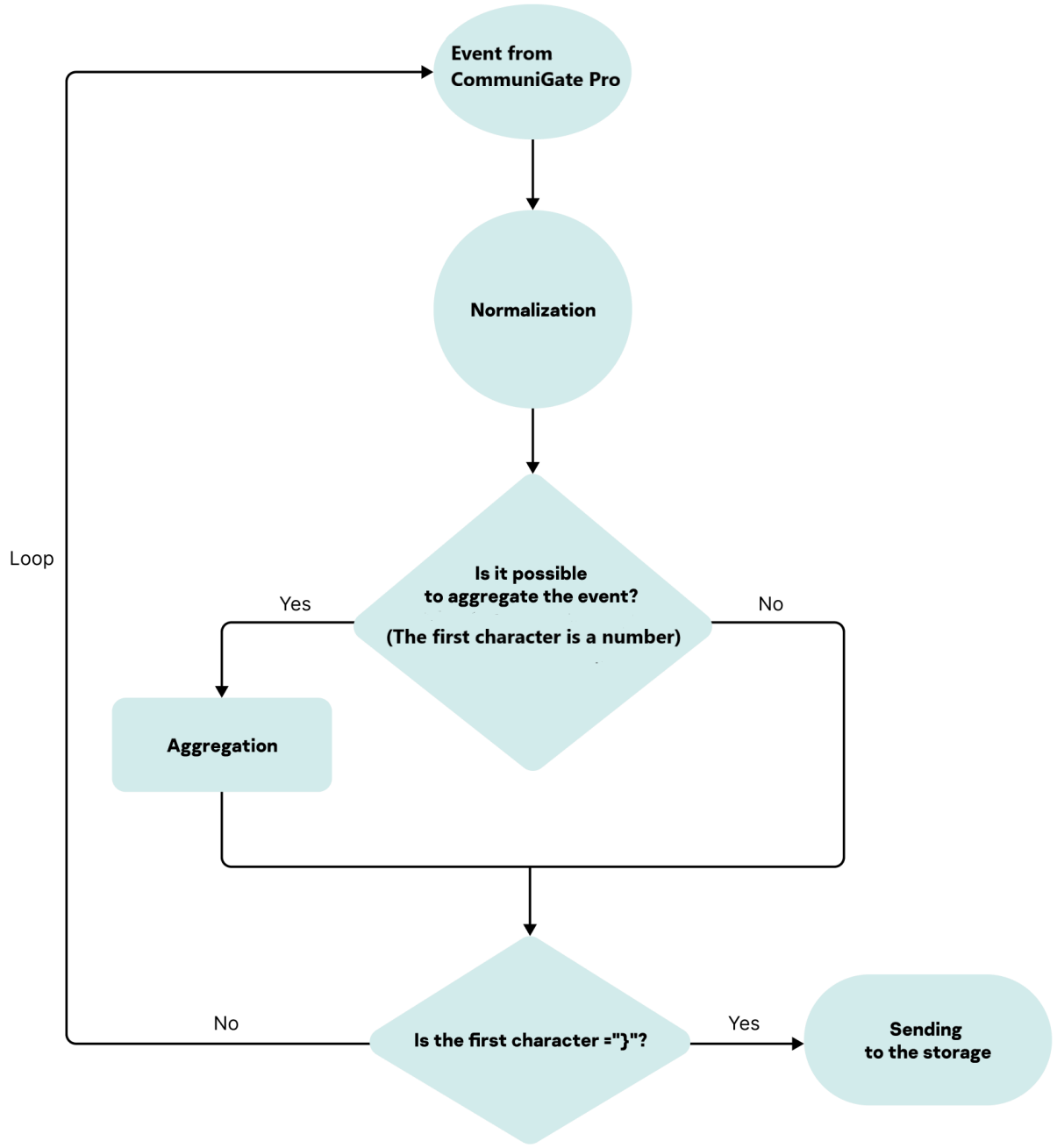
4. Normalization of the aggregated event

   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.



Processing diagram for events from the CommuniGate Pro system

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 web 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**.

   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**.

   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 web interface. 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 involves the following steps:

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. Select the type of access: **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 scree, 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 lowercase 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 scree, 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 lowercase 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 in which you want to mount the bucket 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 the 'kuma' user in the operating system.

<kuma_gid> is the ID of the 'kuma' group 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 https://learn.microsoft.com/en-us/office/office-365-management-api/get-started-with-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 **Event parsing** 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 KUMA 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 KUMA 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 KUMA 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 KUMA 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. On the **Transport** tab, you also need to specify the port and protocol that are specified in the kuma-kont-config.yaml configuration file in the KUMA Collector group of settings.

   2. Installing a collector in the KUMA network infrastructure.

2. Creating a user account in MS SQL.

   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 KUMA 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 KUMA 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.
   - KUMA 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 KUMA 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 KUMA 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 KUMA collector

If the TCP protocol is used to send events, when the connection to the KUMA 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.

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.
`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 KUMA 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 KUMA collector. Only works on first run or when the state file does not exit. The default value is 0.

Setting	Description
`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 KUMA 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 KUMA collector. Only works on first run or when the state file does not exit. The default value is 0.
`address`	Required setting in the **KUMA Collector** group. Fully qualified domain name (FQDN) or IP address of the server where the KUMA collector is installed.
`port`	Required setting in the **KUMA Collector** group. Port of the KUMA collector.
`protocol`	Required setting in the **KUMA Collector** group. Data transfer protocol used by the KUMA collector. If the udp protocol is used, events are not resent if connection to the KUMA collector is lost. Available values: • **tcp** • **udp**

Configuring receipt of VK WorkSpace Mail events

KUMA allows monitoring VK WorkSpace Mail 1.23 events received in syslog format.

Configuring event receiving consists of the following steps:

1. Configuring the export of VK WorkSpace Mail events
2. Creating a KUMA collector for VK WorkSpace Mail events.

   To receive VK WorkSpace Mail audit events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] VK WorkSpace Mail syslog** normalizer. Also, on the **Transport** tab, you need to specify the port and protocol in accordance with steps 2.2, 2.3 of the *Configuring the export of VK WorkSpace Mail events* instructions.
3. Installing a collector in the KUMA network infrastructure.
4. Verifying receipt of VK WorkSpace Mail events in the KUMA collector.

   You can verify that the VK WorkSpace Mail event source server is correctly configured in the Searching for related events section of the KUMA web interface.

Configuring the export of VK WorkSpace Mail events

To configure the export of events to KUMA:

1. Go to the **Settings for duplicating user actions to external storage** menu.
2. Select the **Duplicate user actions to syslog** check box.
   a. In the **Syslog server address** field, enter the IP address or FQDN of the KUMA collector.
   b. In the **Syslog server port** field, specify the port that the KUMA collector is listening on.
   c. In **Syslog protocol, TCP or UDP** field, select the communication protocol.
   d. In the **Syslog data ID** field, specify bein.

The export of VK WorkSpace Mail events is configured.

Configuring receipt of Angie events

KUMA allows monitoring audit events of the Angie or Angie Pro web server (version 1.10.1 or later) deployed on Linux devices.

Configuring the sending and receipt of events involves the following steps:

1. Configuring the export of Angie events
2. Creating a KUMA collector for Angie events

   To receive events, in the KUMA Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Angie syslog** normalizer.
3. Installing a collector in a KUMA network infrastructure
4. **Verifying receipt of Angie events in the KUMA collector**

   To verify whether receipt of events from Angie and Angie Pro is configured correctly, you can search for related events in the KUMA web interface.

Configuring the export of Angie events

To configure the export of Angie events:

1. Check the configuration file for any syntax errors by running the following command:
   ```
   sudo angie -t
   ```
   Possible messages when checking the configuration file

   If there are no syntax errors in the configuration file, Angie displays the following notifications:

   ```
   angie: the configuration file /etc/angie/angie.conf syntax is ok
   ```

```
angie: configuration file /etc/angie/angie.conf test is successful
```

If any syntax errors are found in the configuration file, Angie tells you what the error is, for example:

```
angie: [emerg] "http" directive is not allowed here in etc/angie/http.d/angie-to-
siem.conf:1
angie: configuration file /etc/angie/angie.conf test failed
```

For more information about configuration file syntax, refer to the official documentation of Angie.

2. If the syntax check is successful (the result is `test is successful`), back up the original configuration file using the following command:
   ```
   sudo cp /etc/angie/angie.conf /etc/angie/angie_backup.conf
   ```

3. Open the *angie.conf* configuration file by running the following command:
   ```
   sudo vi /etc/angie/angie.conf
   ```
   By default, the standard angie.conf configuration file is stored in the `/etc/angie` directory.

   If a non-standard configuration file is used in the operation of the Angie web server, you must open that file. You can locate this file using the following command:
   ```
   sudo angie -V
   ```
   The location of the file is specified in the `--config-path` option.

4. Make sure that in the `http` context, the `include` directive is specified, which allows managing configuration using child configuration files, by running the following command:
   ```
   include /etc/angie/http.d/*.conf
   ```

5. Create the -to-siem.conf configuration file in the `/etc/angie/http.d/` directory by running the following command:
   ```
   touch /etc/angie/http.d/angie-to-siem.conf
   ```

6. Open the angie-to-siem.conf configuration file by running the `sudo vi /etc/angie/http.d/angie-to-siem.conf` command.

7. In the `log_format` directive, specify the format of the events to be sent as follows:
   ```
   log_format syslog
   device_event_category=access_log|time=$time_iso8601|version=$angie_version|http_metho
   ```

8. In the `access_log` directive, specify the method to be used to send event information via Syslog:
   ```
   access_log syslog:server=<collector_IP>:<port> syslog
   ```

9. If you also need to send information about events with errors, add a line with the `error_log` directive:
   ```
   error_log syslog:server=<collector_IP>:<port>
   ```
   The message format from the `error_log` directive cannot be changed; the message is always sent in the following form:

   ```
   YYYY/MM/DD HH:MM:SS [LEVEL] PID#TID: *CID MESSAGE
   ```

   where:

   - `YYYY/MM/DD` is the date in the year/month/day format.
   - `HH:MM:SS` is the time in the hours:minutes:seconds format.
   - `[LEVEL]` is the error severity level.

- `PID#TID` is the process ID (PID) and thread ID (TID).
- `*CID` is the unique connection number (Connection ID).
- `MESSAGE` is the text of the error or warning message.

10. Check the syntax of the configuration file for errors using the `sudo angie -t` command, as described in step 1 earlier in this section.
11. Reload the configuration by running the following command:
    ```
    sudo service angie reload
    ```

The export of events from Angie is configured.

Configuring receipt of Dallas Lock 8 events

KUMA allows monitoring events of the Dallas Lock 8 version 11.4.2 system received in CEF format.

Configuring event receiving consists of the following steps:

1. **Configuring the export of Dallas Lock 8 events**
2. **Creating a KUMA collector for Dallas Lock 8 events.**

   When configuring the receipt of Dallas Lock 8 audit events using the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Confident Dallas Lock syslog CEF** normalizer. Also, on the **Transport** tab, you need to specify the port and protocol in accordance with steps 2.a and 2.b of the Configuring the export of Dallas Lock 8 events instructions.
3. **Installing a collector in a KUMA network infrastructure**
4. **Verifying receipt of Dallas Lock 8 events in the KUMA collector**

   You can verify that the Dallas Lock 8 event source server is configured correctly by searching for related events in the KUMA web interface, in the **Resources and services** → **Active services** section.

Configuring the export of Dallas Lock 8 events

To configure the sending of events to KUMA:

1. In Dallas Lock 8, go to **Download logs** menu.
2. Select the **Export logs to SIEM system** check box and do the following:
   a. In the **Server** field, enter the IP address or FQDN of the KUMA collector.
   b. In the **Port** field, specify the port that the KUMA collector is listening on.
   c. Select the **CEF** download format.
   d. Select **UTF-8** download encoding.
3. Select the check boxes next to the types of events that you want to send to KUMA.

4. In the **Log download period** drop-down list, select **1 minute**.

5. Click **OK**.

Event export is configured.

Configuring receipt of Dallas Lock Unified Control Center events

KUMA allows monitoring events of the Dallas Lock Unified Control Center 4.0 (hereinafter referred to as UCC) received in the CEF format.

Configuring event receiving consists of the following steps:

1. Configuring the export of Dallas Lock UCC events
2. Creating a KUMA collector for Dallas Lock UCC events

   When configuring the receipt of Dallas Lock UCC audit events using the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Confident Dallas Lock syslog CEF** normalizer. Also, on the **Transport** tab, you must specify the port and protocol in accordance with steps 2.c, 2.d of the Configuring the export of Dallas Lock UCC events instructions.
3. Installing a collector in a KUMA network infrastructure
4. **Verifying receipt of Dallas Lock UCC events in the KUMA collector**

   You can verify that the Dallas Lock UCC event source server is configured correctly by searching for related events in the KUMA web interface, in the **Resources and services** → **Active services** section.

Configuring the export of Dallas Lock UCC events

To configure the export of Dallas Lock UCC events to KUMA:

1. In the Dallas Lock UCC, go to the **Export security incidents** menu.
2. Select the **Export security incidents to SIEM system** check box and do the following:
   a. In the **Format** drop-down list, select the **CEF** format.
   b. In the **Server** field, enter the IP address or FQDN of the KUMA collector.
   c. In the **Protocol** field, specify the communication protocol.
   d. In the **Port** field, specify the port that the KUMA collector is listening on.
3. Click **Apply**.

Event export is configured.

Configuring receipt of ProFTPD events

KUMA allows monitoring ProFTPD events on Linux devices. Events are collected and audited by ProFTPD, after which the events are sent to KUMA via the rsyslog syslog server.

Configuring the receipt of ProFTPD events involves the following steps:

1. Configuring audit of ProFTPD events.
2. Creating a KUMA collector for ProFTPD events

   When creating a collector in the KUMA web interface, at the **Transport** step, you need to specify the port and protocol configured for the syslog server to send events. To receive audit events from ProFTPD, at the **Event parsing** step, select the **[OOTB] Proftpd syslog** normalizer.
3. Installing a collector in a KUMA network infrastructure
4. Configuring a syslog server to send events.
5. **Verifying receipt of ProFTPD events by the KUMA collector**

   To verify whether receipt of events from ProFTPD is configured correctly, you can search for related events in the KUMA web interface.

Configuring audit of ProFTPD events.

To configure event audit in the ProFTPD system:

1. Create a backup copy of the /etc/proftpd/proftpd.conf configuration file.
2. Open the proftpd.conf file and add or change the following settings:

   ```
   LogFormat default "%t %a %A %b %F %H %h %l %u \"%r\" %s %{file-modified} %{gid} %{transfer-status}"
   LogFormat auth "%t %a %A %b %F %H %h %l %u \"%r\" %s %{file-modified} %{gid} %{transfer-status}"
   ExtendedLog /var/log/proftpd/access.log ALL default
   ExtendedLog /var/log/proftpd/auth.log AUTH auth
   ```
3. Save changes and restart the service by running the following command:

   ```
   sudo systemctl restart proftpd
   ```

Audit of ProFTPD events is configured.

ProFTPD does not perform log rotation. To reduce the risk of disk space filling up, we recommend using an external log rotation tool such as Logrotate or Logadm.

Configuring a syslog server to send ProFTPD events

The rsyslog service is used to transmit events from the server to the KUMA collector.

To configure the sending of events from the server where the ProFTPD system is installed 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 commands:

   ```
   yum install rsyslog
   systemctl enable rsyslog.service
   systemctl start rsyslog.service
   ```

2. In the /etc/rsyslog.d directory, create the proftpd.conf file with the following content, depending on your protocol:

   - To send events over TCP:

     ```
     $ModLoad imfile
     $InputFileName /var/log/proftpd/*.log
     $InputFileTag tag_proftpd_log:
     $InputFileStateFile proftpd_log
     $InputRunFileMonitor
     if $syslogtag contains 'tag_proftpd_log' then {
     *.* action(type="omfwd" target="@<IP address of the KUMA collector>" port="<port of the KUMA collector>" protocol="tcp")
     stop
     }
     ```

   - To send events over UDP:

     ```
     $ModLoad imfile
     $InputFileName /var/log/proftpd/*.log
     $InputFileTag tag_proftpd_log:
     $InputFileStateFile proftpd_log
     $InputRunFileMonitor
     if $syslogtag contains 'tag_proftpd_log' then {
     *.* action(type="omfwd" target="@<IP address of the KUMA collector>" port="<port of the KUMA collector>" protocol="udp")
     stop
     }
     ```

3. Save the changes to the proftpd.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 Vsftpd events

KUMA allows monitoring Vsftpd events on Linux devices. Events are collected and audited by Vsftpd, after which the events are sent to KUMA via the rsyslog syslog server.

Configuring the receipt of Vsftpd events involves the following steps:

1. Configuring audit of Vsftpd events.
2. Creating a KUMA collector for Vsftpd events

   When creating a collector in the KUMA web interface, at the **Transport** step, you need to specify the port and protocol configured for the syslog server to send events. To receive audit events from Vsftpd, at the **Event parsing** step, select the **[OOTB] Vsftpd syslog** normalizer.
3. Installing a collector in a KUMA network infrastructure
4. Configuring a syslog server to send events.
5. **Verifying receipt of Vsftpd events by the KUMA collector**

   To verify whether receipt of events from Vsftpd is configured correctly, you can search for related events in the KUMA web interface.

Configuring audit of Vsftpd events.

To configure event audit in the Vsftpd system:

1. Create a backup copy of the /etc/vsftpd/vsftpd.conf configuration file.
2. Open the vsftpd.conf file and add or change the following settings:

   ```
   xferlog_enable=YES
   xferlog_std_format=NO
   xferlog_file=/var/log/vsftpd.log
   log_ftp_protocol=YES
   ```
3. Save changes and restart the service by running the following command:

   ```
   sudo systemctl restart vsftpd
   ```

Audit of Vsftpd events is configured.

Vsftpd does not perform log rotation. To reduce the risk of disk space filling up, we recommend using an external log rotation tool such as Logrotate or Logadm.

Configuring a syslog server to send Vsftpd events

The rsyslog service is used to transmit events from the server to the KUMA collector.

To configure the export of events from the server where the Vsftpd system is installed 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 commands:

   ```
   yum install rsyslog
   systemctl enable rsyslog.service
   systemctl start rsyslog.service
   ```

2. In the /etc/rsyslog.d directory, create the vsftpd.conf file with the following content, depending on your protocol:

   - To send events over TCP:

     ```
     $ModLoad imfile
     $InputFileName /var/log/vsftpd.log
     $InputFileTag tag_vsftpd_log:
     $InputFileStateFile vsftpd_log
     $InputRunFileMonitor
     *.* @@<IP address of the KUMA collector>:<port of the KUMA collector>
     ```

     For example:

     ```
     *.* @@192.1.3.4:5858
     ```

   - To send events over UDP:

     ```
     $ModLoad imfile
     $InputFileName /var/log/vsftpd.log
     $InputFileTag tag_vsftpd_log:
     $InputFileStateFile vsftpd_log
     $InputRunFileMonitor
     *.* @<IP address of the KUMA collector>:<port of the KUMA collector>
     ```

     For example:

     ```
     *.* @192.1.3.4:5858
     ```

3. Save the changes to the vsftpd.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 HAProxy events

KUMA allows monitoring HAProxy (version 2.4) load balancer events in HTTP, TCP, HTTP CLF, Default, HTTPS (for version 2.6 and later), TCP CLF (for version 3.2 and later) formats received via the syslog protocol.

Configuring the receipt of HAProxy events involves the following steps:

1. Configuring export of HAProxy events
2. Creating a KUMA collector for HAProxy events

   To receive HAProxy events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] HAProxy syslog** normalizer, and at the **Transport** step, select the **udp** connector type.
3. Installing a collector in a KUMA network infrastructure
4. **Verifying receipt of HAProxy events by the KUMA collector**

   You can verify that the receipt of events from HAProxy is configured correctly by searching for related events in the KUMA web interface.

Configuring export of HAProxy events

To configure the export of HAProxy events:

1. Check the configuration file for any syntax errors by running the following command:

   `haproxy -c -V -f /etc/haproxy/haproxy.cfg`

   Possible messages when checking the configuration file

   If there are no syntax errors in the configuration file, HAProxy displays a notification `Configuration file is valid`.

   If any syntax errors are found in the configuration file, HAProxy tells you what the error is, for example:

   ```
   [NOTICE] (284146): haproxy version is 2.4.24-0ubuntu0.22.04.2
   ```

   ```
   [NOTICE] (284146): path to executable is /usr/sbin/haproxy
   ```

   ```
   [ALERT] (284146): parsing [/etc/haproxy/haproxy.cfg:25] : unknown option 'httpdlog'; did you mean 'httplog' maybe ?
   ```

   ```
   [ALERT] (284146): Error(s) found in configuration file : /etc/haproxy/haproxy.cfg
   ```

   ```
   [ALERT] (284146): Fatal errors found in configuration
   ```

   For more information about configuration file syntax, refer to the official documentation of HAProxy.

2. If the syntax check is successful, back up the original configuration file using the following command:

   `sudo cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy_backup.cfg`

3. Open the haproxy.cfg configuration file by running the following command:

   `sudo vi /etc/haproxy/haproxy.cfg`

   The next steps of the instructions assume that the same logging settings are applied to all groups of servers. In this case, the logging settings are specified in the `global` section, and in the `defaults`, the `log global` line is used.

If you need a more granular logging configuration, you need to remove the `log global` line from the `defaults` section. Because the `frontend`, `backend`, and `listen` sections inherit parameters from the `defaults` section by default, this disables global logging. In this case, you need to specify logging settings in each section by adding a line of the following form:

```
log <IP_address_of_KUMA_collector> [len <length_in_bytes>] [format
<syslog_message_format>] [sample <ranges>:<sample_size>] <facility> [<level>
[<minimum_level>]]
```

4. In the `global` section, specify the IP address of the KUMA collector:

   ```
   log <collector_IP_address>:<port>
   ```

5. In the `global` section, add the following line to add the host name to the syslog header:

   ```
   log-send-hostname <hostname>
   ```

6. In the `defaults` section, add a line to use global logging settings:

   ```
   log global
   ```

7. In the `defaults` section, specify the logging format:

   - For layer 4 balancers (TCP):

     ```
     option tcplog
     ```

     Example of a defaults section with settings

     ```
     defaults
     log global
     mode tcp
     option tcplog
     option dontlognull
     timeout connect 5000
     timeout client 50000
     timeout server 50000
     errorfile 400 /etc/haproxy/errors/400.http
     errorfile 403 /etc/haproxy/errors/403.http
     errorfile 408 /etc/haproxy/errors/408.http
     errorfile 500 /etc/haproxy/errors/500.http
     errorfile 502 /etc/haproxy/errors/502.http
     errorfile 503 /etc/haproxy/errors/503.http
     errorfile 504 /etc/haproxy/errors/504.http
     ```

   - For layer 7 balancers (HTTP):

     ```
     option httplog
     ```

     Example of a defaults section with settings

     ```
     defaults
     log global
     mode http
     option httplog
     option dontlognull
     timeout connect 5000
     timeout client 50000
     timeout server 50000
     errorfile 400 /etc/haproxy/errors/400.http
     errorfile 403 /etc/haproxy/errors/403.http
     errorfile 408 /etc/haproxy/errors/408.http
     errorfile 500 /etc/haproxy/errors/500.http
     errorfile 502 /etc/haproxy/errors/502.http
     errorfile 503 /etc/haproxy/errors/503.http
     errorfile 504 /etc/haproxy/errors/504.http
     ```

8. If you want to enable logging of HTTP request headers, add the following line to the `frontend` or `listen` section:

   `capture request header <header_name> len <header_length_in_bytes>`

   Each header must be specified on a separate line, for example:

   ```
   capture request header User-agent len 50
   capture request header Host len 30
   capture request header X-Forwarded-For len 15
   ```

9. If you want to enable logging of HTTP response headers, add the following line to the `frontend` or `listen` section:

   `capture response header <header_name> len <header_length_in_bytes>`

   Each header must be specified on a separate line, for example:

   ```
   capture response header User-agent len 50
   capture response header Host len 30
   capture response header X-Forwarded-For len 15
   ```

10. If you want to enable logging of HTTP-cookie, add the following line to the `frontend` or `listen` section:

    `capture cookie <cookie_name> len <cookie_size_in_bytes>`

    Only the first cookie file with the specified name is logged.

11. Check the configuration file for syntax errors using the `haproxy -c -V -f /etc/haproxy/haproxy.cfg` command, as described in step 1 earlier in this article.

12. Reload the configuration by running the following command:

    ```
    sudo systemctl reload haproxy
    ```

13. Verify that the HAProxy load balancer has started and is operating normally by running the following command:

    ```
    sudo systemctl status haproxy
    ```

Event export from HAProxy is configured.

Configuring receipt of Veeam Backup & Replication events

KUMA allows you to monitor the events of the Veeam Backup & Replication system (version 12.1) received via the syslog protocol.

Configuring the receipt of Veeam Backup & Replication events involves the following steps:

1. Configuring the export of Veeam Backup & Replication events
2. Creating a KUMA collector for Veeam Backup & Replication events

   When creating a collector in the KUMA web interface, at the **Transport** step, you need to specify the type of connector that is used when configuring event export. To receive Veeam Backup & Replication events, at the **Event parsing** step, select the **[OOTB] Veeam Software Veeam Backup & Replication syslog** normalizer.

3. Installing a collector in a KUMA network infrastructure

4. **Verifying receipt of Veeam Backup & Replication events by the KUMA collector**

   You can verify that the receipt of events from Veeam Backup & Replication is configured correctly by searching for related events in the KUMA web interface.

Configuring the export of Veeam Backup & Replication events

Event export must be configure as under an account with administrator rights.

To configure the export of Veeam Backup & Replication events via syslog:

1. In the Veeam Backup & Replication web interface, open the **Options** → **Event forwarding** section.
2. In the **Syslog servers** section, click **Add**.
3. In the **Server** field, specify the fully qualified domain name (FQDN) or IPv4 address of the KUMA collector server. You cannot specify an IPv6 address in this field.
4. In the **Port** field, specify the port that the KUMA collector is listening on.

   By default, the Veeam Backup & Replication system sends events to the following ports, depending on the protocol used:

   - 514 for TCP and UDP
   - 6514 for TLS
5. In the **Transport** drop-down list, select a protocol: **TCP**, **UDP**, or **TLS**.
6. Click **OK**.

Event forwarding from Veeam Backup & Replication to the KUMA collector is configured.

Configuring receipt of Apache Tomcat events

KUMA allows monitoring audit events of an Apache Tomcat web server version 9.0 and later, 10.1 and later, and 11.0 and later deployed on Linux devices. Events are collected and audited by Apache Tomcat, after which the events are sent to KUMA via the rsyslog syslog server.

Configuring Apache Tomcat event reception involves the following steps:

1. Configuring Apache Tomcat event audit
2. Creating a KUMA collector for Apache Tomcat events.

   To receive Apache Tomcat events, in the KUMA Collector Installation Wizard, at the **Event parsing** step, select the **Apache Tomcat [OOTB]** normalizer.
3. Installing a collector in a KUMA network infrastructure
4. Configuring a syslog server to send events.

5. **Verifying receipt of Apache Tomcat events in the KUMA collector.**

   You can verify that the receipt of events from Apache Tomcat is configured correctly by searching for related events in the KUMA web interface.

Configuring Apache Tomcat event audit

To configure event audit in the Apache Tomcat system:

1. Connect to the Apache Tomcat server via SSH using an account with administrator rights.

   > Further steps assume that the standard environment variables `$CATALINA_HOME` and `$JAVA_HOME` are configured on the server.

   For more information on variables, see the Apache Tomcat installation guide.

2. Back up the original configuration file using the following command:

   ```
   sudo cp $CATALINA_HOME/conf/server.xml $CATALINA_HOME/conf/server_backup.xml
   ```

3. Open the Apache Tomcat configuration file for editing using the following command:

   ```
   sudo vi $CATALINA_HOME/conf/server.xml
   ```

4. Find the element `<Valve className="org.apache.catalina.valves.AccessLogValve" ... />` and replace it as follows to use `JsonAccessLogValve`:

   ```
   <Valve className="org.apache.catalina.valves.JsonAccessLogValve" directory="logs"
     prefix="access_log" suffix=".txt"
     maxDays="7"
     pattern="%h %a %t %m %U %q %r %H %A %v %p %s %b %X %I %{User-Agent}i %{Cookie}c"/>
   ```

   If using `JsonAccessLogValve` is not possible, keep `AccessLogValve`, but set up an extended template to collect more fields:

   ```
   pattern="host=%h|remoteAddr=%a|time=%t|method=%m|path=%U|query=%q|request=%r|protocol=
   Agent=%{User-Agent}i|Cookie=%{Cookie}c"
   ```

   Example configuration with `AccessLogValve`:

   ```
   <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
     prefix="access_log" suffix=".txt"
     maxDays="7"
     pattern="host=%h|remoteAddr=%a|time=%t|method=%m|path=%U|query=%q|request=%r|protocc
     Agent=%{User-Agent}i|Cookie=%{Cookie}c"/>
   ```

   The normalizer also supports the Common and Combined templates for `AccessLogValve` and `JsonAccessLogValve`.

5. Restart the Apache Tomcat web server using the following commands:

   ```
   sudo $CATALINA_HOME/bin/shutdown.sh
   sudo $CATALINA_HOME/bin/startup.sh
   sudo systemctl restart tomcat.service
   ```

Audit of Apache Tomcat events is configured.

Configuring the syslog server to send Apache Tomcat events.

The rsyslog service is used to transmit events from the server to the KUMA collector.

To configure the sending of events from the server where the Apache Tomcat system is installed to the collector:

1. Make sure that the rsyslog service is installed on the server that is the event source. For this purpose, execute the following command:

   ```
   systemctl status rsyslog.service
   ```

   If the service is running, a notification similar to the following is displayed:

   ```
   Active: active (running) since Tue 2025-09-23 15:31:30 MSK; 1 week 0 days ago
   ```

   If the rsyslog service is not installed on the server, install it by executing the following commands:

   ```
   sudo apt install rsyslog
   ```

   ```
   sudo systemctl enable rsyslog.service
   ```

   ```
   sudo systemctl start rsyslog.service
   ```

2. Check the syntax of the configuration file by running the following command:

   ```
   rsyslogd -N1
   ```

   Possible messages when checking the configuration file

   If there are no syntax errors in the configuration file, rsyslog displays the following notifications:

   ```
   rsyslogd: version 8.2112.0, config validation run (level 1), master config
   /etc/rsyslog.conf
   rsyslogd: End of config validation run. Bye.
   ```

   If any syntax errors are found in the configuration file, the rsyslog service tells you what the error is, for example:

   ```
   rsyslogd: version 8.2112.0, config validation run (level 1), master config
   /etc/rsyslog.conf
   rsyslogd: module 'imfile' already in this config, cannot be added  [v8.2112.0 try
   https://www.rsyslog.com/e/2221 ]
   rsyslogd: module 'imfile' already in this config, cannot be added  [v8.2112.0 try
   https://www.rsyslog.com/e/2221 ]
   rsyslogd: parameter "maxLinesAtOnce" not permited in inotify mode - ignored [v8.2112.0
   try https://www.rsyslog.com/e/2222 ]
   rsyslogd: $WorkDirectory: /var/lib/rsyslog not a directory - directive ignored
   [v8.2112.0 try https://www.rsyslog.com/e/2181 ]
   rsyslogd: imfile: no working or state file directory set, imfile will create state
   files in the current working directory (probably the root dir). Use
   global(workDirectory="/some/path") to set the working directory [v8.2112.0 try
   https://www.rsyslog.com/e/2450]
   ```

3. Open the rsyslog.conf configuration file by running the following command:

   ```
   sudo vi /etc/rsyslog.conf
   ```

4. Confirm that the file includes a line that allows loading configurations from the /etc/rsyslog.d directory by running the following commands:

   ```
   # Include all config files in /etc/rsyslog.d/
   ```

   ```
   include(file="/etc/rsyslog.d/*.conf" mode="optional")
   ```

5. Create the apache_tomcat_to_siem.conf configuration file in the /etc/rsyslog.d/ directory by running the following command:

   ```
   sudo touch /etc/rsyslog.d/apache_tomcat_to_siem.conf
   ```

6. Open the created configuration file apache_tomcat_to_siem.conf using the following command:

   ```
   sudo vi /etc/rsyslog.d/apache_tomcat_to_siem.conf
   ```

7. Run the following command to find out the absolute path to the Apache Tomcat directory:

   ```
   printenv CATALINA_HOME
   ```

   > You need to know the absolute path to specify it in the configuration file because the rsyslog service does not let you use environment variables.

8. Add the rules for reading the `Access Log` and for sending data to the KUMA collector to the configuration file:

   ```
   input(
           type="imfile"
           File="<absolute_path_from_step_7>/logs/access_log*.txt"
           Tag="apachetomcat"
   )
   if $programname == "apachetomcat" then {
           action(
                   type="omfwd"
                   target="<FQDN_или_IP-KUMA_collector_address"
                   port="<KUMA_collector_port"
                   protocol="udp_or_tcp"
           )
   }
   ```

9. Check the syntax of the configuration file for errors using the `sudo rsyslogd -N1` command, as described in step 2.

10. Restart the rsyslog service by running the following command:

    ```
    sudo systemctl restart rsyslog
    ```

11. Verify that the rsyslog service has started and is operating normally by running the following command:

    ```
    sudo systemctl status rsyslog
    ```

The sending of events from the server where the Apache Tomcat system is installed to the collector is configured.

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
+	+		+	+	Source 4 identified
+		+	+	+	Source 5 identified
+	+	+	+	+	Source 6 identified
	+	+		+	Source not identified
	+		+	+	Source not identified

DeviceProduct	DeviceHostName	DeviceAddress	DeviceProcessName	TenantID (determined automatically)	
		+	+	+	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 web interface 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 KUMA web interface, go to the **Source status** section.
2. On the **List of event sources** tab in the menu ⋮, click **Settings**.
3. This opens the **Settings of event source detection** window; in that window, in the **Grouping fields for determining the source** 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 **Grouping fields for determining the source** 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 web interface 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 KUMA web interface, go to the **Source status** section.
2. This opens the **List of event sources** window; in that window, 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 solution web interface in the **Source status** section 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 web interface under **Source status** 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

Event sources are displayed in the table in the **Source status** → **List of event sources** section. 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. To search for a single character, enter that character and press **Enter**. 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

In the **Source status** → **List of event sources** section, 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. To find a policy by searching for a single character, enter that character and press **Enter**.

  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 a source, sources with an event count are displayed in the **Flow** field. If only monitoring policies of the **byEPS** type or no monitoring policies are assigned to a source, sources with an EPS value are displayed in the **Flow** field.

  When sorting by the **Stream** column, sources with events per second (**byEPS**) come first, followed by sources with an event count (**byCount**).

  After receiving the user's request, KUMA selects event sources and sorts them by the values in the **Flow** column at the time of receiving the request. After that, the values in this column are updated in accordance with the user's request.

The displayed value may not match the sorting order because event data in event sources is dynamically updated to provide up-to-date information.

- **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 the expected behavior.
- You can click the **Build 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 as a CSV file

You can download information about one or more event sources and the monitoring policies applied to them as a CSV file (comma-separated) 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 file contains the following parameters: **Status**, **Name**, **Monitoring policy**, **Lower limit**, **Upper limit**, **Stream**, **Tenant**. Identical fields for identifying sources are saved in the CSV file in the order specified by the user when configuring the identification of sources. When downloading data for multiple sources, the order of sources configured in the KUMA web interface is not preserved. If you want to sort the list of sources, you can do so in Excel or use grep with sorting.

To download event source information to a CSV file:

1. In the KUMA web console, in the **Source status** → **List of event sources** section, 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 multiple event sources in the table by selecting the check box in the header of the first column of the table and selecting one of the following values:

   - **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. However, you cannot clear the check box for an individual event source.
   - **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. After selecting all sources on the page, you can clear the check boxes for individual event sources.

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 located in the ⋮ menu.

   An 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_YYYYMMDD_HHMMSS.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.

You can view the graph for a single event source in the KUMA web console in the **Source status** → **List of event sources** section by clicking 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 KUMA web console, in the **Source status** → **List of event sources** section, 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 **Build chart** button in the upper part of the table.

   > Depending on the size of your browser window, the **Build 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.

You can use the **Search** field to look for monitoring policies. The search will be carried out in the following columns: **Name**, **Tenant**, **Type**, **Schedule** (name of the day and time). To find a policy by searching for a single character, enter that character and press **Enter**.

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 **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 **List of event sources** section, 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 policies of different types) 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 **List of event sources** page.

   - 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 KUMA web interface, under **Source status** → **Monitoring policies**, click **Add policy** and configure the monitoring policy in the displayed window:

2. In the **Policy name** field, enter a name for the policy you are creating. The name must contain from 1 to 128 Unicode characters and must be unique within the selected tenant.

   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 **Set up a schedule by days of the week** toggle switch.

      b. In the **Days of 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 **Set up a schedule for specific days** 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 **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**
  - **Schedule**

    Schedule changes are applied within a minute, when KUMA checks for policies that need to be enabled or disabled.
- **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**

    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 **Date the policy was 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 KUMA web console, in the **Source status** → **List of event sources** section, 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. However, you cannot clear the check box for an individual event source.

   - **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. After selecting all sources on the page, you can clear the check boxes for individual event 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 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 or sorting is not available for the **Sources**, **Schedule**, **Policy update status**, **Date the policy was 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 KUMA web console, in the **Source status** → **List of event sources** section, 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 or sorting is not available for the **Sources**, **Schedule**, **Policy update status**, **Date the policy was 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** 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 you enable the policy manually**.

   The default selection is **For**, 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 KUMA web interface, in the **Source status** → **Monitoring policies** section, 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). To find a policy by searching for a single character, enter that character and press **Enter**.

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 KUMA web interface, in the **Source status** → **Monitoring policies** section, 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). To find a policy by searching for a single character, enter that character and press **`Enter`**.

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.

Managing assets

Assets represent devices and systems in the organization for which security events are received. These systems and devices can include workstations, servers, network equipment, mobile devices, and other infrastructure objects. After adding assets to KUMA, their IDs are added to enriched events, and, when analyzing events, you get additional information about your organization's computers.

You can view information about assets, search for assets by specified criteria, edit or delete assets, and export asset data to a CSV file.

Asset categories

You can group assets in categories, which you can use when configuring 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, KUMA asset categories have the following severity levels: 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. If you specified a relative period and selected the frequency of categorization, for example, hourly, every time categorization starts, the condition will consider asset information that is up-to-date at the time of starting the categorization.

  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. Under **Conditions**, 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, for example, `CVE-2017-0143`.
CVSS	>, >=, =, <=,<	Severity level of CVE vulnerabilities on the asset. The CVSS parameter takes values from 0 to 10. Not applicable to vulnerabilities from Kaspersky Security Center.

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 Kaspersky Security Center 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, check 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, check 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, check 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, check 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, check 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, the **Assets for given conditions** window containing a list of assets that satisfy the search conditions will be displayed.

- **Reactive**—When a correlation rule is triggered, the asset is moved to the specified group.

In KUMA, 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 of the KUMA web interface. 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 update of assets in the selected category. During the update, new assets can be linked or previously linked assets can be unlinked. 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. Open the **Assets** section in the KUMA web interface.
2. Open the category creation window by doing the following:
   a. In the list of tenants, expand the tenant to which you want to add the new category.
   b. Hover over the parent category and do one of the following:
      - Click the plus (+) icon.
      - In the menu (⋮) select **Add subcategory**.

   The **Add category** details area appears in the right part of the web interface 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.
   - If necessary, change the category in the **Parent** drop-down list.
   - The **Tenant** field displays the tenant whose structure contains your selected parent category. You cannot change the previously selected tenant.
   - If necessary, change the severity of the category in the **Severity** drop-down list. The default value is **Low**.
   - If necessary, in the **Description** field, you can add a note consisting of up to 4000 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 are periodically assigned to a category if they satisfy the specified condition. If an asset no longer satisfies this condition, such an asset is unlinked from the category, regardless of whether the assets were added automatically or manually.

     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. Under **Conditions**, 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, for example, `CVE-2017-0143`.
CVSS	>, >=, =, <=,<	Severity level of CVE vulnerabilities on the asset. The CVSS parameter takes values from 0 to 10. Not applicable to vulnerabilities from Kaspersky Security Center.

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 Kaspersky Security Center 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, check 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, check 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, check 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, check 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, check 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, the **Assets for given conditions** window containing a list of assets that satisfy the search conditions will be displayed.

- **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 KUMA, 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. Open the **Assets** section in the KUMA web interface.

2. Click the gear ( ) 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:

   - **FQDN**
   - **IP address**
   - **Asset source**
   - **Owner**
   - **MAC address**
   - **Created by**
   - **Updated**
   - **Tenant**
   - **CII category**
   - **Archived**
   - **Status**
   - **Score ML**

   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

KUMA 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 data to a CSV file.

Simple search

To find an asset using simple search:

1. In the **Assets** section of the KUMA web interface, 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 **Assets** section of the KUMA web interface, 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 to vulnerabilities from Kaspersky Security Center.

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	• Kaspersky Security Center. • 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. Name of the Kaspersky Security Center administration group in which the asset is placed.
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 KUMA web interface, select **Assets** → select a category with the relevant assets → select an asset.
- In the KUMA web interface, select **Alerts** → click the link with the relevant alert → select the asset in the **Related endpoints** section.
- In the KUMA web interface, select **Events** → 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 KUMA retain the names that were assigned to them at the source. You can change these names in the KUMA web interface.

- **Tenant**—the name of the tenant that owns the asset.
- **Asset source**—source of information about the asset. There may be several sources. For instance, information can be added in the KUMA web interface or by using the API, or it can be imported from Kaspersky Security Center, KICS/KATA, and MaxPatrol reports.

  When using multiple sources to add information about the same asset to KUMA, you should take into account the rules for merging asset data.
- **Created**—date and time when the asset was added to KUMA.
- **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 KUMA, 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 Kaspersky Security Center.

    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 Kaspersky Security Center.
  - **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 Kaspersky Security Center. This information is available for the assets imported from Kaspersky Security Center.
  - **Host ID**—ID of the Kaspersky Security Center *Network Agent* from which the asset information was received. This information is available for the assets imported from Kaspersky Security Center. This ID is used to determine the uniqueness of the asset in Kaspersky Security Center.
  - **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 asset custom fields.
- 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 Kaspersky Security Center.
- **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 Kaspersky Security Center 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 Kaspersky Security Center 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 Kaspersky Security Center.

The tasks are available when integrated with Kaspersky Security Center and when integrated with Kaspersky Endpoint Detection and Response.

Adding assets

You can add asset information to KUMA in the following ways:

- Manually.

  You can add an asset using the KUMA web interface or the API. 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.

- Import assets.

  You can import assets from Kaspersky Security Center, KICS/KATA, and MaxPatrol reports.

When adding assets, assets that already exist in KUMA can be merged with the assets being added.

Asset merging algorithm:

1. Checking the uniqueness of assets in Kaspersky Security Center or KICS/KATA 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/KATA is determined by the combination of the **IP address**, **KICS/KATA server IP address**, and **KICS/KATA 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**, **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**, **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.

Examples of asset field comparison are displayed here.

Information about assets can be generated from various sources. If the added asset and the KUMA 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 KUMA. 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/KATA.
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/KATA. • 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/KATA. • 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.

Field name	Update procedure
Software info	Data from KICS/KATA 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 the KUMA web interface.

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 the KUMA web interface is used to edit asset information that was received from Kaspersky Security Center or KICS/KATA, you can edit the following asset information:

- Name.
- Category.

If asset information was added manually, you can edit the following asset data when editing these assets in the KUMA web interface:

- 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.

Merging duplicate assets

As a result of importing asset information from different sources, the list of assets in KUMA may contain duplicates of the same assets. You can contact Technical Support to get an utility that lets you analyze the list of assets and merge duplicate assets.

Adding asset information in the KUMA web interface

To add an asset in the KUMA web interface:

1. In the **Assets** section of the KUMA web interface, click the **Add asset** button.

   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 tree ( ) 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 plus ( ) and minus ( ) 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 and asset vulnerability information from Kaspersky Security Center

All assets that are protected by Kaspersky Security Center are registered in it. You can import into KUMA the information about assets or vulnerabilities of assets that Kaspersky Security Center protects. To do so, you need to configure integration between the applications in advance.

In Kaspersky Security Center integration settings, you can configure the frequency of automatic import of information about assets, and, if necessary, import assets manually. Importing assets manually does not affect the time of the next scheduled import. From the Kaspersky Security Center database, KUMA 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.

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

By default, basic asset information is imported every hour, and information about asset attributes is imported every 12 hours. Attribute information is imported only for existing assets, not for new or deleted assets.

If a large number of assets are registered in Kaspersky Security Center (tens of thousands hosts), the import task may take a significant amount of time to complete (up to 15–20 hours). The following errors may be encountered during the import process:

```
Kaspersky Security Center assets not fetched. Post "https://<KSC
address>:13299/api/v1.0/login": read tcp <KUMA address>:57498-><KSC address>:13299:
read: connection reset by peer
```

```
Kaspersky Security Center assets not fetched. list child: Resource is unavailable
```

```
Kaspersky Security Center assets not fetched. Post "https://<KSC
address>:13299/api/v1.0/login": EOF
```

To help the import complete sooner and avoid such errors, you need to additionally manage integration settings.

Additional configuration of the integration

For KUMA 3.2 and later, you can specify the number of hosts to be processed per query when importing vulnerability information.

To set the number of hosts:

1. Create a vuln_iterator_config.json file with the following contents:

   ```
   {
   "ksc1.domain.ru": k,
   "ksc2.domain.ru": k
   }
   ```

   where:

   - `ksc1.domain.ru, ksc2.domain.ru` are addresses of Kaspersky Security Center servers specified in the KUMA integration settings.
   - `k` sets the number of hosts per query for which information about vulnerabilities is requested. The maximum number of vulnerabilities per query is 10,000.

2. Place the created file:

   - For a single-server KUMA installation, in /opt/kaspersky/kuma
   - For a distributed KUMA installation on multiple servers, run the following command on any cluster controller:

     ```
     export POD=$(k0s kubectl get pods --namespace kuma -l "app=core" -o jsonpath="
     {.items[0].metadata.name}");
     k0s kubectl cp --no-preserve -c core ./vuln_iterator_config.json
     kuma/$POD:/opt/kaspersky/kuma/vuln_iterator_config.json
     ```

   Changes take effect the next time the import task is run.

During the next import of information from Kaspersky Security Center, if KUMA does not receive data about a previously added asset, such an asset is marked as archived. If no information about this asset is received within the next 30 days, the asset is deleted.

KUMA does not import Kaspersky Security Center assets with the **Last visible** parameter greater than 30 days.

If Kaspersky Security Center encounters errors while running the import tasks, KUMA displays such errors. If basic asset information is not available in KUMA during the import of asset attribute information (for example, if the assets were deleted during the import), the task completes without errors, but the attribute information for these assets is not loaded.

KUMA provides the following ways of importing information about assets or asset vulnerabilities from KSC:

- Importing asset information and asset vulnerability information for assets of all KSC Servers.
- **Importing asset information and asset vulnerability information for assets of an individual KSC Server.**

  To import information about assets or asset vulnerabilities of an individual Kaspersky Security Center Server:

  1. In the KUMA web interface, go to the **Settings** → **Kaspersky Security Center** section.

     The **Kaspersky Security Center integration by tenant** window opens.

  2. Select the tenant for which you want to perform the import.

     The **Kaspersky Security Center integration** window opens.

  3. In the **Connections** table, click the connection to the selected Kaspersky Security Center Server.

     This opens the connection settings window.

  4. If the Kaspersky Security Center Server has secondary servers or groups, and you do not want to import information about their assets or vulnerabilities:

     a. Click the **Load hierarchy** button.

     b. Clear the check boxes next to the secondary Servers or groups for which you do not want to perform the import. By default, the check boxes are selected.

     c. If you want to import asset information or asset vulnerability information from new groups, select the **Import assets from new groups** check box. This check box is cleared by default.

     d. Click the **Save** button.

5. Import asset information and asset vulnerability information for assets in one of the following ways:

   - If you want to import information about the basic parameters of assets (protection status, versions of anti-virus databases, hardware), click the **Import KSC assets** button.
   - If you want to import information about other parameters of assets (vulnerabilities, software, owners), click the **Import KSC assets attributes**.

   Information about the assets or asset vulnerabilities of the selected Kaspersky Security Center Server is imported into KUMA.

Importing asset information from MaxPatrol

You can import asset information from the MaxPatrol system into KUMA.

You can use the following import arrangements:

- Importing from reports about scan results of network devices of the MaxPatrol 8 system.

  The import is performed through the API using the maxpatrol-tool on the server where the KUMA Core is installed. The tool is included in the KUMA distribution kit and is located in the installer archive in the /kuma-ansible-installer/roles/kuma/files directory.

- Importing data from MaxPatrol VM.

  Data is imported via the API using the kuma_pvtm utility. The tool is included in the KUMA distribution kit and is located in the installer archive in the /kuma-ansible-installer/roles/kuma/files directory.

Imported assets are displayed in the KUMA web interface in the **Assets** section. If necessary, you can edit the settings of assets.

Importing data from MaxPatrol reports

Importing asset information form a report is supported for MaxPatrol 8.

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:

   - General administrator, Tenant administrator, Tier 2 analyst, or Tier 1 analyst role.
   - Access to the tenant into which the assets will be imported.

- Permissions for using API requests GET /api/v3/users/whoami and POST /api/v3/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>`

   You can download the Core certificate in the KUMA web interface.

   Example: `./maxpatrol-tool --kuma-rest kuma.example.com:7223 --token token.txt --tenant Main example.xml --cert /tmp/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 utility 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, `kuma.example.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 account for which you are generating an API token must have the General administrator, Tenant administrator, Tier 2 administrator, or Tier 1 administrator role.
`--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`
`--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.

Examples:

- `./maxpatrol-tool --kuma-rest kuma.example.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.

Error message	Description
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 KUMA 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 KUMA API. Data import is supported for MaxPatrol VM versions from 1.1 to 2.7 inclusive.

Configuring the import of asset information from MaxPatrol VM to KUMA involves the following steps:

1. Preparing KUMA and MaxPatrol VM.

   You must create user accounts and a KUMA token for API operations.

2. Creating a configuration file with data export and import settings.

3. Importing asset data into KUMA 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.

When re-importing existing assets, assets that already exist in KUMA 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 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 imports this data as one software, despite the different software installation paths in the asset.

If you set strict_import to false in the kuma-ptvm-config.yaml configuration file, data from MaxPatrol is imported for all assets, including those without fields that KUMA requires and those with field values that KUMA does not allow. If the FQDN of an asset contains a space or underscore ("_"), data for such assets is not imported into KUMA, 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 KUMA and in MaxPatrol VM with the minimum necessary set of permissions to use API requests.
2. Create user accounts for which you will lager generate an API token.

   Requirements imposed on accounts for which the API token is generated:

   - General 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/api/v1/assets/import.
3. Generate a token for access to the KUMA REST API.

Creating the configuration file

To create the configuration file:

1. Go to the KUMA installer folder by executing the following command:

   ```
   cd kuma-ansible-installer/roles/kuma/files/
   ```
2. Extract the kuma-ptvm.tar.gz archive by running the following command:

   ```
   tar -xvf kuma-ptvm.tar.gz
   ```
3. 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
   ```
4. Edit the settings in the kuma-ptvm-config.yaml configuration file.
5. Save the changes to the file.

The configuration file will be created. Go to the Importing asset data step.

Importing asset data

To import asset information:

1. If you want to import asset information from MaxPatrol VM into KUMA 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, follow these steps:

   a. Run the kuma-ptvm utility with the following options:

   ```
   ./kuma-ptvm --config <path to the kuma-ptvm-config.yaml file> --download
   ```

If necessary, you can use the `--cvss-version <CVSS version number>` option to specify the version of CVSS for which you want to load asset vulnerability values. Available values:

- `2` loads only CVSS version 2 values.
- `3` loads only CVSS version 3 values. If a vulnerability is described only for version 2, the CVSS field is empty.
- `latest` (default) loads the values of the latest available version of CVSS. If the vulnerability does not have a CVSS version 3 value, the version 2 value is used.

Examples:

```
./kuma-ptvm --config kuma-ptvm-config.yaml --cvss-version 2 --download
./kuma-ptvm --config kuma-ptvm-config.yaml --cvss-version 3 --download
./kuma-ptvm --config kuma-ptvm-config.yaml --cvss-version latest --download
```

For each tenant specified in the configuration file, a separate file is created with a name of the form <KUMA tenant ID>.JSON. Also, during export, a 'tenants' file is created, containing a list of JSON files to be uploaded to KUMA. 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.

c. Import asset information into KUMA:

```
./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. 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.

Solution.

- 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.

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.

Description of settings in the kuma-ptvm-config.yaml configuration file

Setting	Description	Values
`log_level`	Optional setting in the `General settings` section. Logging level.	Available values: • `trace` • `info` • `warning` • `error` Default setting: `info`.
`period`	Optional setting in the `General settings` section. 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` section. 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 KUMA.
`parallel_downloading`	Optional setting in the `General settings` section. Sets the downloading mode when importing data from MaxPatrol.	Available values: • `true` downloads CSV files in parallel. • `false` downloads CSV files sequentially to reduce the load on MaxPatrol. The default setting is `true`. We recommend specifying `false` if you need to reduce the load on MaxPatrol during data import.
`endpoint`	Required setting in the `KUMA settings` section. URL of the KUMA API server. For example, kuma.example.com:7223	–
`token`	Required setting in the `KUMA settings` section. KUMA API token.	–
`ignore_server_cert`	Optional setting in the `KUMA settings` section. Validation of the KUMA certificate.	Available values: • `true` to disable KUMA certificate validation. • `false` to enable KUMA 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 settings` section. URL of the MaxPatrol API server.	–
`user`	Required setting in the `MaxPatrol VM settings` section. MaxPatrol API user name.	–
`password`	Required setting in the `MaxPatrol VM settings` section. MaxPatrol API user password.	–
`secret`	Required setting in the `MaxPatrol VM settings` section. MaxPatrol API secret.	–

Setting	Description	Values
`ignore_server_cert`	Optional setting in the `MaxPatrol VM settings` section. Validation of the MaxPatrol certificate.	Available values: • `true` to disable the validation of the MaxPatrol certificate. • `true` 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` section. 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` section. 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` section. Tenant ID in KUMA. 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` section. Regular expression for searching the FQDN of an asset.	–
`networks`	Optional setting in the `Tenant map` section. One or more subnets.	–
`default_tenant`	Optional setting. The default KUMA tenant for data about assets that could not be allocated to tenants specified in the 'Tenants' section.	–

Importing asset information from KICS for Networks

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 start a task to update KICS/KATA asset information for a tenant:

1. In the KUMA web interface, open the **Settings** → **KICS/KATA** section.
2. Select the relevant tenant.

   This opens the **KICS/KATA server integration** window.
3. Click the **Import assets** button.

A task to receive account data from the selected tenant is added to the **Task manager** section of the KUMA web interface.

Examples of asset field comparison during import

Each imported asset is compared to the matching KUMA 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 KUMA 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 KUMA asset: the FQDN and IP fields are filled in and match. The assets are merged.
- Imported asset 3 and KUMA 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 KUMA 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 KUMA 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 KUMA 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
KUMA 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 KUMA 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 KUMA 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 KUMA asset: no matching fields. The assets are not merged.
- Imported asset 4 and KUMA asset: no matching fields. The assets are not merged.

Assigning a category to an asset

To assign a category to one asset:

1. In the KUMA web interface, go to the **Assets** 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 KUMA web interface, 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 **Assets** section of the KUMA web interface, select the check box in the heading 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 are unlinked from the selected category.

Editing the parameters of assets

In KUMA, 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 **Assets** section of the KUMA web interface, click the asset that you want to edit.

   The **Asset details** area opens in the right part of the window.

2. Click the **Edit** button.

   The **Edit asset** window opens.

3. 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/KATA.
   - **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.

4. Assign or change the category of the asset:

   a. Click the tree ( ) 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.

5. Click the **Save** button.

Asset parameters have been changed.

Archiving assets

In KUMA, the archival functionality is available for the following types of assets:

- For assets imported from Kaspersky Security Center and KICS.

  If KUMA 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 KUMA receives information about the asset from the source before the retention period for archived assets expires.

- Combined assets

  When importing, KUMA 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:

- KUMA did not receive information about the asset from Kaspersky Security Center or KICS/KATA.
- 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 **TTL for archived assets (days)**.

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/KATA.
- Connection with the Kaspersky Security Center server has not been established for more than 30 days.

Archived assets that participate in dynamic categorization remain archived. An archived asset can have its CII category assigned or changed. If such an asset ends up in an alert or incident, the CII category of the alert or incident also changes, which may affect the visibility of the alert or incident for users with restricted CII access.

To configure the archived assets retention period:

1. In the KUMA web interface, select the **Settings** → **Assets** section.

   This opens the **Assets** window.

2. 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.

3. 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 KUMA web interface, 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

If you no longer need to receive information from an asset or information about the asset has not been updated for a long time, you can have KUMA delete the asset. Deletion can be performed by the General administrator, Tenant administrator, Tier 2 analysts, and Tier 1 analysts. If an asset was deleted, but KUMA once again begins receiving information about that asset from Kaspersky Security Center, KUMA recreates the asset with a new ID.

In KUMA, you can delete assets in the following ways:

- Automatically.

  KUMA automatically deletes only archived assets. KUMA deletes an archived asset if the information about the asset has not been updated for longer than the retention period of archived assets.

- Manually.

To delete an asset manually:

1. In KUMA web interface, in the **Assets** section, click the asset that you want to delete.

   This opens the **Asset information** window in the right-hand part of the web interface.

2. Click the **Delete** button.

   A confirmation window opens.

3. Click **OK**.

The asset is deleted and no longer appears in the alert or incident card.

Bulk deletion of assets

In the KUMA 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 **Assets** section of the KUMA web interface, select the category that contains the relevant assets.

   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 KUMA.

To install updates and fix vulnerabilities in third-party applications on an asset in KUMA:

1. Open the asset details window in one of the following ways:
   - In the KUMA web interface, select **Assets** → select a category with the relevant assets → select an asset.
   - In the KUMA web interface, select **Alerts** → click the link with the relevant alert → select the asset in the **Related endpoints** section.
   - In the KUMA web interface, select **Events** → search and filter events → select the relevant event → 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 the **Accept selected EULAs** button in the KUMA web interface.

   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 KUMA when the hierarchy is loaded during import of assets from Kaspersky Security Center. First, you need to configure KUMA integration with Kaspersky Security Center.

To move an asset to a selected administration group:

1. Open the asset details window in one of the following ways:
   - In the KUMA web interface, select **Assets** → select a category with the relevant assets → select an asset.
   - In the KUMA web interface, select **Alerts** → click the link with the relevant alert → select the asset in the **Related endpoints** section.
2. In the asset details window, click the **Move to KSC group** button.
3. Click the **Move to KSC group** button.
4. Select the group in the opened window.

   The selected group must be owned by the same tenant as the asset.
5. Click the **Save** button.

The selected asset will be moved.

To move multiple assets to a selected administration group:

1. In the KUMA web interface, select the **Assets** section.
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 the **Move to KSC group** button.

   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.

The specific group of an asset is displayed in the asset details.

Kaspersky Security Center assets information is updated in KUMA 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 KUMA. When an attempt is made to move such an asset to an administration group in which it is already located, KUMA returns the **Failed to move assets to another KSC group** error.

Asset audit

KUMA can be configured to generate asset audit events under the following conditions:

- Asset was added to KUMA. 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/KATA.
- 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 KUMA. The application monitors manual deletion of assets, as well as automatic deletion of assets imported from Kaspersky Security Center and KICS/KATA, whose data stopped coming.
- Vulnerability info was added to the asset. The application 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/KATA.
- Asset vulnerability was resolved. The application 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 application monitors the assignment of an asset category to an asset.
- Asset was removed from a category. The application monitors the deletion of an asset from an asset category.

By default, if asset audit is enabled, under the conditions described above, KUMA 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 KUMA web interface, open **Settings** → **Asset audit**.
2. Perform one of the following actions with the tenant for which you want to configure asset audit:
   - Add the tenant by using the **Add tenant** button if this is the first time you are configuring asset audit for the relevant tenant.

     In the opened **Asset audit** window, select a name for the new tenant.
   - Select an existing tenant in the table if asset audit has already been configured for the relevant tenant.

     In the opened **Asset audit** window, the tenant name is already defined and cannot be edited.
   - Clone the settings of an existing tenant to create a copy of the conditions configuration for the tenant for which you are configuring asset audit for the first time. To do so, select the check box next to the tenant whose configuration you need to copy and click **Clone**. In the opened **Asset audit** window, select the name of the tenant to use the copied configuration.
3. For each condition for generating asset audit events, select the destination to where the created events will be sent:
   a. In the group of settings for the relevant type of asset audit events, use the **Add destination** drop-down list to select the type of destination to which you want to send the created events:
      - 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 solution.

      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.
4. 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 KUMA web interface, open the **Settings** → **Asset audit** section and select the tenant for which you want to enable or disable asset audit.

   The **Asset audit** window opens.

2. Select or clear the **Disabled** check box in the upper part of the window.

3. Click **Save**.

   By default, when asset audit is enabled in KUMA, 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 KUMA web interface, open the **Settings** → **Asset audit** section and select the tenant for which you want to enable or disable a condition for generating asset audit events.

   The **Asset audit** window opens.

2. Select or clear the **Disabled** check box next to the relevant conditions.

3. Click **Save**.

   For conditions with the **Disabled** check box selected, only audit events are created, and base events are not created.

Detection of unknown assets

By processing events, KUMA lets you detect unknown assets in the infrastructure that are not included in the list of added KUMA assets. The asset must have an IP address or host name. An unknown asset being detected can be normal or it may indicate an information security incident.

The following predefined correlation rules are used to detect unknown assets:

- [OOTB][Net] Detection unknown assets in device fields.
- [OOTB][Net] Detection unknown assets in source fields.
- [OOTB][Net] Detection unknown assets in destination fields.

These correlation rules use filters to configure the monitored network segment and exclusion filters. You can use filters to customize the rule logic that suits your particular network infrastructure. Triggered rules add detected unknown assets to the "[OOTB][Net] Detection unknown assets" list.

The "[OOTB][Net] Detection unknown assets" active list retains a record about a detected asset for 7 days. If any rule is triggered again, no duplicate record about the previously added asset is added to the active list.

Correlation rules, filters, and active list are predefined. You can also import these resources from the repository.

Correlation rules

[OOTB][Net] Detection unknown assets in device fields

This rule analyzes the data in the DeviceAddress and DeviceHostName fields of events. If the IP address or host name from these fields is not found in the list of KUMA assets, the rule is triggered.

The following conditions must be satisfied for the rule to trigger:

- The asset is not included in the "[OOTB][Net] Exceptions from the device addresses" exclusion filter.
- The IP address belongs to a subnet specified in the "[OOTB][Net] Controlled network device" filter.

[OOTB][Net] Detection unknown assets in source fields

This rule analyzes the data in the SourceAddress and SourceHostName fields of events. If the IP address or host name from these fields is not found in the list of KUMA assets, the rule is triggered.

The following conditions must be satisfied for the rule to trigger:

- The asset is not included in the "[OOTB][Net] Exceptions from the source addresses" exclusion filter.
- The IP address belongs to a subnet specified in the "[OOTB][Net] Controlled network source" filter.

[OOTB][Net] Detection unknown assets in destination fields

This rule analyzes the data in the DestinationAddress and DestinationHostName fields of events. If the IP address or host name from these fields is not found in the list of KUMA assets, the rule is triggered.

The following conditions must be satisfied for the rule to trigger:

- The asset is not included in the "[OOTB][Net] Detection unknown assets in destination fields" exclusion filter.
- The IP address belongs to a subnet specified in the "[OOTB][Net] Detection unknown assets in destination fields" filter.

Filters

Exclusion filters

Exclusion filters contain lists of IP addresses and host names of assets which prevent correlation rules from triggering when detected. Add IP addresses and host names to exclusion filters before using correlation rules. For the rules to work correctly, filters must not be empty.

Exclusion filters include:

- [OOTB][Net] Exceptions from the device addresses.
- [OOTB][Net] Exceptions from the source addresses.
- [OOTB][Net] Exceptions from the destination addresses.

Exclusion filters are located in the OOTB/Integration directory.

Monitored subnet filters

Subnet filters contain lists of subnets which correlation rules monitor for new assets. Add subnets to filters before using correlation rules. Without a list of subnets, correlation rules are never triggered.

Subnet filters include:

- [OOTB][Net] Controlled network device.
- [OOTB][Net] Controlled network source.
- [OOTB][Net] Controlled network destination.

Subnet filters are located in the OOTB/Integration directory.

Configuring the detection of unknown assets in your network

To configure the detection of unknown assets:

1. In the KUMA web interface, go to the **Resources** → **Filters** section.
2. In the list of filters, select the following filters one by one and specify the IP addresses or host names that you want to exclude from correlation rules:
   - [OOTB][Net] Exceptions from the device addresses.
   - [OOTB][Net] Exceptions from the source addresses.
   - [OOTB][Net] Exceptions from the destination addresses.

3. In the list of filters, select each of the following filters one by one and specify the subnets that you want to be monitored for unknown assets:

   - [OOTB][Net] Controlled network device.
   - [OOTB][Net] Controlled network source.
   - [OOTB][Net] Controlled network destination.

   Unknown asset detection is configured for the specified subnets. Information about detected assets will be added to the "[OOTB][Net] Detection unknown assets" active list.

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 the KUMA web interface in the **Settings** → **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 KUMA web interface, in the **Settings** → **Assets** section, click the **Add field** button.

   An empty row is added to the **Custom fields** table. You can add multiple rows with the custom field settings at once.

2. 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.

3. Click **Save**.

   A custom field is added to the asset data model.

To delete or edit a custom asset field:

1. In the KUMA web interface, open **Settings** → **Assets**.
2. Make the necessary changes in the **Custom fields** table:
   - To delete a custom field, click the ✕ icon next to the row with the settings of the required field. 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, drag the rows by the ⣿ icon
3. Click **Save**.

The changes are made.

Critical information infrastructure assets

In KUMA, you can tag assets related to the critical information infrastructure (CII) of the Russian Federation. This allows you to restrict the KUMA 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 KUMA.

General 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 KUMA web interface, in the **Assets** section, select the required asset.

   The **Asset details** window opens.

2. 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 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**.

3. Click **Save**.

Integration with other solutions

In this section, you'll learn how to integrate KUMA with other applications to enrich its functionality.

Integration with Kaspersky Security Center

KUMA can be integrated with Kaspersky Security Center Windows as well as Kaspersky Security Center Linux. In the following instructions, "Kaspersky Security Center" stands for the application that you have chosen for integration.

You can configure integration with selected Kaspersky Security Center servers for one, several, or all KUMA tenants. If Kaspersky Security Center integration is enabled, you can import information about the assets protected by this application, manage assets using tasks, and import events from the Kaspersky Security Center event database.

> First, you need to make sure that the relevant Kaspersky Security Center server allows an incoming connection for the server hosting KUMA.

Configuring KUMA integration with Kaspersky Security Center includes the following steps:

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.

   The secret (account role in Kaspersky Security Center) for integrating KUMA with Kaspersky Security Center must be created with consideration of how the hierarchy of the Administration Server is organized (availability of virtual servers, server administration features, etc) and types of devices that the Administration Server will manage (OS, type: servers, mobile devices, etc). All these nuances are regulated and configured on the Kaspersky Security Center side.

The following actions can be performed in KUMA on assets from Kaspersky Security Center:

- Starting a task of the **Refreshing** type.
- Starting a task of the **Virus Scan** type.
- Moving assets between Kaspersky Security Center groups.
- Accepting software updates (to fix a vulnerability of an asset in Kaspersky Security Center).

To be able to perform the actions listed above, you can use a predefined account in Kaspersky Security Center with the Main Administrator role. In this case, you do not need to add permissions manually.

You can also use the "Kaspersky Endpoint Security Administrator" predefined role in Kaspersky Security Center, but in that case, you must additionally grant access to the following functionality:

1. Management of administration groups
2. Vulnerability and patch management

   Some additional permissions may be required depending on the configuration of Kaspersky Security Center.

Minimum permissions for integration with Kaspersky Security Center:

- "Access objects regardless of their ACLs" allows you to import Kaspersky Security Center assets into KUMA.

- "Management of administration groups" allows you to move assets between groups in Kaspersky Security Center from the KUMA interface.

- "Basic functionality" allows you to create and run tasks on Kaspersky Endpoint Security hosts.

For more details about creating a user account and assigning permissions to a user, please refer to the *Kaspersky Security Center Help Guide*.

2. **Adding the created user account to the list of exclusions for two-factor authentication**

   In the Kaspersky Security Center Administration Console, exclude the user account created for the integration of KUMA with Kaspersky Security Center from two-factor authentication in Kaspersky Security Center Windows or Kaspersky Security Center Linux.

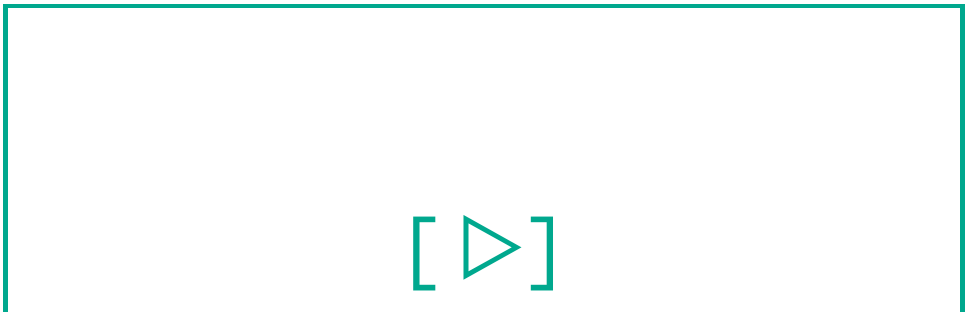
3. **Creating a secret of the credentials type for connecting to Kaspersky Security Center**

4. **Configuring Kaspersky Security Center integration settings**

5. **Creating a connection to the Kaspersky Security Center server for importing information about assets**

   If you want to import information about assets registered on Kaspersky Security Center servers into KUMA, you need to create a separate connection to each Kaspersky Security Center server for each selected tenant.

   If integration is disabled for the tenant or there is no connection to Kaspersky Security Center, an error is displayed in the KUMA web interface when attempting to import information about assets. In this case, the import process does not start.



https://www.youtube.com/watch?v=-MYOlw0sPiY

Configuring Kaspersky Security Center integration settings

To configure the settings for integration with Kaspersky Security Center:

1. Open the KUMA web interface and select the **Settings** → **Integrations** → **Kaspersky Security Center** section.

   The **Kaspersky Security Center integration by tenant** window opens.

2. Select the tenant for which you want to configure integration with Kaspersky Security Center.

   The **Kaspersky Security Center integration** window opens.

3. Use the toggle switch in the **Status** column to enable or disable Kaspersky Security Center integration for the tenant.

   By default, Kaspersky Security Center integration for tenants is disabled.

4. Specify intervals for automatic import of asset information and asset vulnerability information from Kaspersky Security Center:

   a. Click the name of the tenant in the **Tenant** column.

   b. A sidebar is displayed; on this sidebar, in the **KSC assets, hardware information** field, enter the interval in hours for the automatic import of information about the basic attributes of assets (protection status, anti-virus database version, hardware). and must be an integer. The default setting is `1` (1 hour).

   c. In the **KSC assets attributes (vulnerabilities, software, owners)** field, enter the interval in hours for automatic import of information about other attributes of assets (vulnerabilities, software, owners). and must be an integer. The default setting is `12` (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.

   If necessary, you can manually import asset information and asset vulnerability information from Kaspersky Security Center.

5. Click the **Save** button.

The Kaspersky Security Center integration settings for the selected tenant will be configured.

If the tenant you need is missing from the list of tenants, you need to add the tenant to the list of tenants.

Adding a tenant to the list for Kaspersky Security Center integration

To add a tenant to the list of tenants for integration with Kaspersky Security Center:

1. Open the KUMA web interface and select the **Settings** → **Integrations** → **Kaspersky Security Center** section.

   The **Kaspersky Security Center integration by tenant** window opens.

2. Click the **Add settings for a new tenant** button.

3. The **Kaspersky Security Center** sidebar is displayed; on this sidebar, in the **Tenant** drop-down list, select the tenant that you want to add.

4. Click **Add**.

The selected tenant will be added to the list of tenants for integration with Kaspersky Security Center.

Removing a tenant from the list of tenants for Kaspersky Security Center integration

To remove a tenant from the list of tenants for Kaspersky Security Center integration:

1. Open the KUMA web interface and select the **Settings** → **Integrations** → **Kaspersky Security Center** section.

   The **Kaspersky Security Center integration by tenant** window opens.

2. Select the check box for the tenant that you want to delete.

3. Click the **Delete** button and confirm deletion.

The selected tenant is removed from the list of tenants for integration with Kaspersky Security Center.

Creating Kaspersky Security Center connection

To create a new Kaspersky Security Center connection:

1. Open the KUMA web interface and select the **Settings** → **Integrations** → **Kaspersky Security Center** section.

   The **Kaspersky Security Center integration by tenant** window opens.

2. Click the name of the tenant for which you want to create a Kaspersky Security Center connection.

3. A sidebar is displayed; on this sidebar, under **Connections**, click the **New** button and specify the following settings:

   - **Status** is the status of the connection to the selected Kaspersky Security Center server. You can use the toggle switch to enable or disable the connection. By default, the connection to the selected server is inactive. If this is the case, you cannot use this connection to connect to the Kaspersky Security Center server.

   - **Name** (required)—the name of the connection. The name can contain 1 to 128 Unicode characters.

- **URL** (required) is the URL of the Kaspersky Security Center server. The default port is 13299.
- In the **Secret** drop-down list, select the secret with the Kaspersky Security Center account credentials or **create a new secret**.

  1. Click the plus (+) icon.

     The secret window is displayed.

  2. 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 Kaspersky Security Center account credentials.

     c. In the **Type** drop-down list, select **credentials**.

     d. In the **User** and **Password** fields, enter the account credentials for your Kaspersky Security Center server.

     e. If you want, enter a **Description** of the secret.

  3. Click **Save**.

  You can change the selected secret by clicking the icon.

4. If you want KUMA to import only assets that are connected to secondary servers or included in groups:

   a. Make sure that the connection to the Kaspersky Security Center server is enabled.

   b. In the **Hierarchy** section, click the **Upload** button.

   c. Select the servers and groups from which you want to import asset information.

   d. If you want to import assets only from new groups, select the **Import assets from new groups** check box.

5. If you want KUMA to import information about the assets of all Kaspersky Security Center servers:

   a. Make sure that the connection to the Kaspersky Security Center server is enabled.

   b. If you want to import information about the basic parameters of assets, click the **Import KSC assets** button.

   c. If you want to import information about other parameters of assets, click the **Import KSC assets attributes** button.

6. Click **Save**.

The connection to the Kaspersky Security Center server is now created. It can be used to import information about assets from Kaspersky Security Center to KUMA and to create asset-related tasks in Kaspersky Security Center from KUMA.

Editing Kaspersky Security Center connection

To edit a Kaspersky Security Center connection:

1. Open the KUMA web interface and select the **Settings** → **Integrations** → **Kaspersky Security Center** section.
   The **Kaspersky Security Center integration by tenant** window opens.
2. Select the name of the tenant for which you want to configure Kaspersky Security Center integration.
   This opens the **Kaspersky Security Center** window.
3. Click the name of the Kaspersky Security Center connection you want to edit.
   The window with the selected Kaspersky Security Center connection parameters opens.
4. Make the necessary changes to the settings.
5. Click the **Save** button.

The Kaspersky Security Center connection will be changed.

Deleting Kaspersky Security Center connection

To delete a Kaspersky Security Center connection:

1. Open the KUMA web interface and select the **Settings** → **Integrations** → **Kaspersky Security Center** section.
   The **Kaspersky Security Center integration by tenant** window opens.
2. Select the name of the tenant for which you want to configure Kaspersky Security Center integration.
   This opens the **Kaspersky Security Center** window.
3. Select the Kaspersky Security Center connection that you want to delete.
4. Click the **Delete** button.

The Kaspersky Security Center connection will be deleted.

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 [OOTB] KSC MSSQL, [OOTB] KSC MySQL, or [OOTB] KSC PostgreSQL connector.
- 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 the KUMA web interface, in the **Resources** section, under **Resources**, select **Connectors**.

   b. Find the relevant predefined connector in the folder structure, select the check box next to this connector and click **Duplicate**.

   c. 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())
```

d. In the **Connection** section, in the **Database kind** drop-down list, select the type of database you are using.

e. If necessary, select the **Secret separately** check box and enter the secret in the **Secret** field.

f. Place the cursor in the **URL** field and in the displayed list, click ✎ in the line of the secret that you are using.

g. 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**.

h. 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 **Create**.

2. Install the collector in the web interface:

a. Start the Collector Installation Wizard in one of the following ways:

- In the KUMA web interface, in the **Resources** section, under **Services configuration**, click **Add event source**.
- In the KUMA web interface, in the **Resources** section, under **Services configuration**, select **Collectors** and click **Add**.

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 flowcharts** tab, click **Add event parsing**.

e. This opens the **Basic event parsing** window; in that window, on the **Normalization scheme** tab, in the **Normalizer** drop-down list, select **[OOTB] KSC from SQL** 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.



https://www.youtube.com/watch?v=ikOEyl3fa18

Kaspersky Endpoint Detection and Response integration

Starting with KATA 8.0, integration with Kaspersky Endpoint Detection and Response and the import of Kaspersky Endpoint Detection and Response telemetry into KUMA are no longer supported. Response through KEDR is not available, attempting to initiate response for an asset causes an error. Only the import of platform action events and alert events is available.

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 4.0 or later 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.

To manage Kaspersky Endpoint Detection and Response actions on Kaspersky Security Center assets, the fully qualified domain name (FQDN) must be specified for the asset in the system (the "Full device name" field). Otherwise, no response actions can be started.

You can also import events to KUMA and receive information about Kaspersky Endpoint Detection and Response alerts (for more details, see the *Configuring integration with an SIEM system* section of the Kaspersky Anti Targeted Attack Platform online 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 application component administrator menu is displayed.
3. In the application 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 `10000` value.

> 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 `/usr/bin/apt-start-sedr-iptables` configuration file, add the `10000` value to 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 KUMA:

1. On the KUMA 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`—for Linux.

2. In the KUMA web interface, 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 the KUMA web interface, 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 application component administrator menu is displayed.
3. In the application 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 <mask>
   ```

   where:

   <mask> is the network mask for addressing cluster servers.

Preparations for exporting events on the Kaspersky Endpoint Detection and Response side are now complete.

In KUMA:

1. On the KUMA 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`—for Linux.
2. In the KUMA web interface, 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 the KUMA web interface, 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 on the KUMA side. To do this, in KUMA, 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 KUMA to begin receiving events in KUMA.

As a result, the integration is configured and KEDR events start arriving in KUMA.

Creating a collector for receiving events from KEDR

To create a collector for receiving events from KEDR:

1. In KUMA → **Resources** → **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**:

    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, 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 displays the *Bad Request: max_events N is greater than the allowed value* error.

        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 server value **Events fetch timeout** 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 in the **Basic event parsing** window, select the [OOTB] KEDR telemetry normalizer in the drop-down list.

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** → **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 KUMA → **Events** section.

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 a new 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 KUMA event details

You can configure the creation of an alert or each received Kaspersky Endpoint Detection and Response 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 KUMA web interface 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.
4. Adding a collector, taking into account the following:
   - At the **Transport** step, specify the connector type (TCP or UDP) and the connection port (must be the same as the port configured in the SIEM connection settings in Kaspersky Anti Targeted Attack Platform).
   - At the **Parsing** step, when selecting a normalizer, you need to specify mappings for KEDR event fields or select the `[OOTB] KATA` normalizer.
   - After adding the correlator, in the collector, you need to specify a destination pointing to the correlator.

You can use a pre-configured correlation rule. In this case configuring the display of a link in the KUMA web interface 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.

To add a category:

1. In the KUMA web interface, select the **Assets** section.
2. On the **All assets** tab, expand the list of tenant categories by doing one of the following:
   - Under **All tenants**, click the menu icon (•••) and select **Expand all**.
   - In the **All tenants** object tree, click the arrow icon (▸) next to the name of the tenant.
3. Select a category or subcategory and do one of the following:
4. Click the add icon (+).
5. Click the menu icon (⋮) and select **Add subcategory**.
   The **Add category** sidebar is dispalyed in the right part of the web interface window.
6. Define the category settings:
   a. In the **Name** field, enter the name of the category.
   b. In the **Parent** drop-down list, select categories from the tree of categories.
   c. In the **Tenant** drop-down list, select a tenant if it is not automatically specified.
   d. 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**.
        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. Under **Conditions**, 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, for example, `CVE-2017-0143`.

CVSS	>, >=, =, <=,<	Severity level of CVE vulnerabilities on the asset. The CVSS parameter takes values from 0 to 10. Not applicable to vulnerabilities from Kaspersky Security Center.

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 Kaspersky Security Center 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, check 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, check 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, check 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, check 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, check 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, the **Assets for given conditions** window containing a list of assets that satisfy the search conditions will be displayed.

- **Reactive**—the category will be filled with assets by using correlation rules.

e. If required, define the values for the following settings:

- Assign a severity to the category in the **Severity** drop-down list.

  The specified severity is assigned to correlation events and alerts associated with the asset.

- Add a description for the category in the **Description** field.

7. Click the **Save** button.

To add an asset:

1. In the KUMA web interface, select the **Assets** section.

2. Click the **Add asset** 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 a name for the asset.

   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. If necessary, you can add one or more server IP addresses.

d. 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.
- In the **Categories** field, select the category of the tenant.
- In the **CII categories** drop-down list, select the critical information infrastructure category of the resource.

4. If necessary, under **Software**, specify the name and version of the operating system.

5. If necessary, in the **Hardware info** section, specify the following parameters of the hardware:

- Name, core count, and clock rate of the CPU in MHz. If necessary, you can click the **Add CPU** button specify more CPUs.
- Free disk space and the total size of the disk in bytes. If necessary, you can click the **Add disk** button specify more disks to be used.
- Name, vendor, and driver version of the network adapter. If necessary, you can click the **Add network card** button to specify more network adapters.
- The clock rate of the RAM in MHz and the total amount of RAM in bytes.

6. Click the **Save** button.

Step 2. Adding a correlation rule

You can create a correlation rule or use the ready-made `[OOTB] KATA Alert` rule.

To add a correlation rule:

1. In the KUMA web interface, select the **Resources** section.

2. Under **Resources**, select **Correlation rules** and click the **Add** button.

3. On the **General** tab, specify the following settings:

a. In the **Name** field, define the rule name.

b. In the **Tenant** drop-down list, select a tenant.

c. In the **Type** drop-down list, select **simple**.

d. In the **Tags** drop-down list, select one or more tags to identify the rule.

e. In the **Propagated fields** field, add the following fields: **DeviceProduct**, **DeviceAddress**, **EventOutcome**, **SourceAssetID**, **DeviceAssetID**

f. 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.
- In the **MITRE Techniques** drop-down list, select the applicable MITRE ATT&CK techniques.

4. On the **Selectors** → **Settings** tab, specify the following filter settings:

a. In the **Filter** drop-down list, select **Create new**.

b. Click **Add group**.

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 fields**.
4. In the **Event fields** 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 fields**.
4. In the **Event fields** field, select **DeviceAssetID**.
5. In the **Operator** field, select **inCategory**.
6. In the **Right operand** field, select **Constant**.
7. Click the **Builder** 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 fields**.

4. In the **Event fields** 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 through 7 under (h) 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. Select the **Output** check box.

b. In the **Enrichment** section, click the **Add enrichment** button.

c. In the **Source kind** drop-down list, select **Template**.

d. In the **Template** field, enter `https://{{.DeviceAddress}}:8443/katap/#/alerts?id={{.EventOutcome}}`.

e. In the **Target field** drop-down list, select **DeviceExternalID**.

> If you want the **DeviceExternalID** value to be displayed as a link in the event properties, make sure that the **Template** field for the `DeviceAddress` parameter contains a URL or fully qualified domain name followed by the "/" character, for example: `https://example.com:8443/` or `https://example.com/`.

f. 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 (**Related events** 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.

> Integration with the CyberTrace web interface requires the CyberTrace TIP Enterprise license.

You can also configure retroscan and event enrichment using CyberTrace. This allows KUMA to get notifications for events in which threat indicators had not been initially detected, but were discovered later after an update of the feeds. This can improve the accuracy of threat detection based on previously logged events.



https://www.youtube.com/watch?v=KMOaROHuwkw

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 KUMA.

   In the enrichment rule, you can specify which data from CyberTrace you want to enrich the event with. We recommend selecting cybertrace-http as the source kind.

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 web interface, 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 application 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 application is installed.

   The wizard starts at step 1, **Welcome to Kaspersky CyberTrace**. You can go to the next step of the wizard by clicking **Next**.

2. At step 2, **Proxy settings**, if your organization uses a proxy server, enter its connection settings. If your organization does not use a proxy server, leave all fields blank.

3. At step 3, **Licensing settings**, select the method for adding a license key for CyberTrace: an activation code or a license key file. Depending on the selected method, specify the activation code or upload a license key file.

4. At step 4, **Service settings**, keep default settings.

5. At step 5, **Data management settings**:

   a. In the **SIEM system** drop-down list, select **KUMA**.

   b. Under **Listen on**, select the **IP and port** option.

   c. In the **IP address** field, enter `0.0.0.0`.

   d. In the **Port** field, enter the port to listen on for events. The default port is `9999`.

   e. Under **Send detection alerts**, in the **IP address** field, enter `127.0.0.1`, and in the **Port** field, enter `9998`.

   Leave the default values for everything else.

6. At step 6, **Certificate settings**, select **Commercial certificate** and add a certificate that allows you to download data feeds from update servers.

7. At step 7, **Feeds settings**, keep default settings.

CyberTrace is configured.

To configure CyberTrace to receive and process requests in the application web interface:

1. In the window of the CyberTrace web interface, switch **Data management** mode: in the left menu, select **System**, and then in the displayed menu, select **General**.

2. Select the **Settings** → **General** section.

3. Under **Listen on**:

   a. Select **IP and port**.

   b. In the **IP address** field, enter `0.0.0.0`.

   c. In the **Port** field, enter the port to listen on for events. The default port is `9999`.

4. Select the **Settings** → **Service alerts** section.

5. In the **Service alert format** field, enter `%Date% alert=%Alert%%RecordContext%`.

6. In the **Records context format** field, enter `|%ParamName%=%ParamValue%`.

7. Select the **Settings** → **Detection alerts** section.

8. In the **Alert format** field, enter
   `Category=%Category%|MatchedIndicator=%MatchedIndicator%%RecordContext%`.

9. On the **Context** tab, in the **Actionable fields** field, enter `%ParamName%:%ParamValue%`.

10. Switch to the **System management** mode: in the left menu, select **General**, then in the displayed menu, select **System**.

11. Select the **Settings** → **Service** section.

12. Under **Web interface**, in the **IP address or host name**, enter `127.0.0.1`.

13. In the upper toolbar, click **Restart service**.

14. Restart the CyberTrace server.

CyberTrace is configured.

Creating event Enrichment rules

To create event enrichment rules:

1. In the KUMA web interface, open **Resources** → **Enrichment rules**.
   The list of available enrichment rules will be displayed.

2. Click the **Add** button to create the new rule.

3. This opens a window; in that window, specify the enrichment rule settings:

   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-http**.

   d. Specify the **URL** of the CyberTrace server to which you want to connect. For example, *example.domain.com:443*.

e. In the **Secret** field, select or create a secret for the connection.

f. In the **Key fields** drop-down list, select the event fields used for enrichment of events with data from CyberTrace.

g. If necessary, use the **Timeout** field to specify the 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.

h. In the **RPS** field, enter the number of requests to the CyberTrace server per second that KUMA can make. The default value is `1000`.

i. In the **Timeout** field, specify the maximum number of seconds KUMA 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.

j. 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.

k. Use the **Major version of Cybertrace** toggle switch to specify the CyberTrace version. Possible values: <5 and ≥ 5. If you select version ≥ 5, requests are executed in the General tenant in CyberTrace and use the new API version corresponding to CyberTrace 5.0 and higher; CyberTrace can also display statistics and retrospective analysis information for enrichment requests from KUMA that are made over HTTP. If you select <5, requests use the Kaspersky CyberTrace 4.x API. The default setting is <5.

l. Use the **Debug** toggle switch to indicate whether or not to enable logging of service operations. Logging is disabled by default.

m. In the **Tags** drop-down list, select the tags for the event enrichment rule.

n. If necessary, in the **Description** field, add up to 4,000 Unicode characters describing the resource.

o. Under **Filter parameters**, 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

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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

4. Click **Create**.

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 information from a data feed by copying value of the **TI indicator** event field from KUMA and searching for it in CyberTrace in the **Indicators** section. All information about the found indicator is displayed on the **Indicator details** page.

Integrating CyberTrace interface

You can integrate the CyberTrace web interface into the KUMA web interface. When this integration is enabled, the KUMA web interface includes a **CyberTrace** section that provides access to the CyberTrace web interface. You can configure the integration in the **Integrations** → **Kaspersky CyberTrace** section of the KUMA web interface.

To integrate the CyberTrace web interface in KUMA:

1. In the KUMA web interface, open the **Resources** → **Resources** → **Secrets**.

The list of available secrets is displayed.

2. Click the **Add** button to create a new secret. This resource is used to store credentials of the CyberTrace server.

   This opens the incident creation window.

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 **Type** 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**.

   The CyberTrace server credentials are now saved and can be used in other KUMA resources.

5. In the KUMA web interface, open **Settings** → **Integrations** → **Kaspersky CyberTrace**.

   The window with CyberTrace integration parameters opens.

6. Manage the settings as follows:

   - Enable the **State** toggle switch if you want to enable the integration of the CyberTrace web interface into the KUMA web interface.
   - **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 a previously created secret of the **credentials** type.
   - 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**.

7. You can configure access to the CyberTrace web interface in the following ways:

   - Use hostname or IP when logging into the KUMA web interface.

     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

     on which the KUMA web interface is deployed.

- Use the FQDN when logging into the KUMA web interface.

  If you are using the Mozilla Firefox browser to manage the solution web interface, the CyberTrace section may fail to display data. In this case, configure the data display (see the instructions below).

8. Click **Save**.

CyberTrace is now integrated with KUMA, and the **CyberTrace** section is displayed in the KUMA web interface.

*To configure the data display in the **CyberTrace** section when using the FQDN to log in to KUMA in Mozilla Firefox:*

1. Clear your browser cache.
2. In the browser's address bar, enter the FQDN of the KUMA web interface with port number 7222 as follows: https://kuma.example.com:7222.

   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 will be created for the URL of the KUMA web interface.
5. In the browser's address bar, enter the URL of the KUMA web interface with port number 7220.
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 the KUMA web interface, 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**.

   A confirmation window opens.
3. If you want to confirm your actions and update the **Internal TI** with data from KUMA events, click **Yes**.

The selected object is now added to the CyberTrace denylist.

Configuring event enrichment and retroscan

Integration of KUMA with Kaspersky CyberTrace (hereinafter referred to as CyberTrace) version 5.0 and later allows enriching events with information about indicators of compromise (IoC) and retrospectively scanning events. Retroscan helps detect indicators that had not been detected at the time of initial enrichment, but appeared later in updated CyberTrace feeds. Retroscan can be used in the following cases:

- The incident has already occurred, but its indicators became known only after the feeds were updated.
- You want to rescan saved events for new threats.

To set up data enrichment and retroscan:

1. Perform the initial configuration in KUMA by following these steps:

   a. Configure CyberTrace integration.

   b. Create two collectors to receive events that you want to enrich with CyberTrace data.

      - To get notifications from CyberTrace about the triggering of indicators, create a collector in which a connector of the **tcp** type is used as the transport, and the normalizer is **[OOTB] CyberTrace**.
      - To configure event enrichment, create a collector with the following parameters:
        - Select any connector (for example, **http**) as the transport.
        - You can choose any normalization method (for example, **json**) with the mapping of the following fields: `SourceHostName`, `DestinationHostName`, `Message`, `DeviceHostName`, `DeviceAddress`, `FileHash`, `OldFileHash`, `RequestUrl`.
        - As the enrichment, create an event enrichment rule in which the data source type is **cybertrace-http**, the CyberTrace version is **≥ 5.0**, and the following normalizer fields are selected in the **Key fields** drop-down list: `SourceHostName`, `DestinationHostName`, `Message`, `DeviceHostName`, `DeviceAddress`, `FileHash`, `OldFileHash`, `RequestUrl`.

2. To configure retroscan in the CyberTrace web interface, follow these steps:

   a. In the **General** → **General** section, enable **API lookup** to save detected indicators and collect statistics a public API request is used.

   b. On the **General** → **Retroscan** page, enable retrospective scan, then select the **Regular expressions** and enable the sources and regular expressions that you want to use for retrospective scan.

   c. In the **General** → **General** section, under **Detection alerts**, specify the IP address and port that CyberTrace will use to send alerts about the detection of indicators of compromise.

   d. In the **General** → **General** section, under **Service alerts**, configure CyberTrace service notifications (for example, notifications about the retroscan task finishing or license key replacement) by specifying the IP address of the collector and the port of the KUMA TCP connector.

   For more information on configuring retroscan, see the *Kaspersky CyberTrace Help*.

Event processing stages during retroscan

Retroscan allows detecting indicators of compromise that had not been known when an event was received, but became available later after an update of Kaspersky CyberTrace feeds.

The retroscan algorithm comprises the following steps:

1. **Sending an event to CyberTrace**

   An event containing indicators is received at the CyberTrace collector. When received, these indicators may not be present in active feeds.

   Example event:

   ```
   {
   "SourceHostName": "45.54.64.52",
   "DestinationHostName": "geardox.site",
   "Message": "5BA7335AD847E5ED6211FB27094B9855",
   "DeviceHostName": "test.domain.ru",
   "DeviceAddress": "5.8.16.77",
   "FileHash": "E1B85D995DC09DE25603D0B7A67D998015F6C045",
   "OldFileHash": "B276263D373A6B56EEAB09673429E1490AEFC5E14738CDF9E1661C36EBD80656",
   "RequestUrl": "https://kaspersky.com/"
   }
   ```

2. **Saving the event in the CyberTrace database**

   If integration over TCP is used, CyberTrace uses regular expressions to extract indicators from the event. If integrated via the REST API (using API lookup), indicators are passed directly in the body of the request. If at the moment when an indicator is processed, the feeds do not contain information that allows to classify it, CyberTrace saves the indicator to the retroscan database for subsequent analysis after feeds are updated.

3. **Running the retroscan task**

   The retroscan task can be scheduled or run manually. To run a retroscan manually, go to the **System** → **Retroscan** page and click **Start now**.

4. **Detecting indicators after a feed update**

   If, during the update process, the feeds deliver indicators that match previously saved events, CyberTrace detects such matches when running the retroscan task.

5. **Generating alerts based on retroscan results**

   After the retroscan is completed, CyberTrace generates alerts about the detection of indicators of compromise and sends these alerts to the KUMA collector.

   Example alert:

   ```
   Category=KUMA_Demo_1_Hash_SHA1|MatchedIndicator=E1B85D995DC09DE25603D0B7A67D998015F6C045|MD
   05:34|geo=bd, eg, in|last_seen=02.02.2021
   11:36|popularity=2|threat=HEUR:Trojan.MSIL.Taskun.gen
   ```

6. **Event processing in KUMA**

   The event is processed by the **[OOTB] CyberTrace** normalizer. The user can match the received notification with the original event for subsequent analysis.

7. **Getting a service alert about the completion of the task**

If CyberTrace is configured to send service alerts, CyberTrace sends an alert about the completion of the retroscan task.

Example alert:

```
Jun 05 16:08:33
alert=KL_ALERT_RetroScanCompleted|iocs_detected=168|iocs_rescanned=1511177|retroscan_report
C2DF-4AED-B430-2E64AEF781FC
```

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, the portal helps KUMA users to make faster and better-informed decisions, providing them with data about URLs, domains, IP addresses, including WHOIS and DNS data.

In the Kaspersky Threat Intelligence Portal Help, you can find information about data provision for the following objects:

- Hashes
- IP addresses
- Domains
- Web addresses

Access to the Kaspersky Threat Intelligence Portal is provided based on a fee. License certificates are created by Kaspersky staff. To obtain a certificate for Kaspersky Threat Intelligence Portal, contact your Kaspersky Technical Account Manager.

Initializing the Kaspersky Threat Intelligence Portal integration

To integrate Kaspersky Threat Intelligence Portal into KUMA:

1. In the KUMA web interface, open the **Resources** section.
2. Under **Resources**, select **Secrets**.

   The list of available secrets will be displayed.
3. Click the **Add** 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.
4. 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.

5. 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.

6. If necessary, select one or more tags for the secret from the **Tags** drop-down list.

7. If you want, enter a **Description** of the secret.

8. Click **Create**.

   The Kaspersky Threat Intelligence Portal account credentials are now saved and can be used in other KUMA resources.

9. Go to the **Settings** → **Integrations** → **Kaspersky Threat Lookup integration**.

10. Use the **Status** toggle switch to enable the integration. The integration is disabled by default.

11. 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 **Resources** → **Secrets** section.

12. If necessary, select a proxy server in the **Proxy** drop-down list.

13. If you want to accept the License Agreement for OSINT data, click the **License Agreement** link. This opens the **License Agreement** window; in that window, select **I accept the License Agreement** and click **Save**. KUMA will start to receive OSINT data after you send a data enrichment request from Kaspersky Threat Intelligence Portal by selecting the **OSINT data** check box in the enrichment settings.

14. Click **Save**.

15. After you save the settings, log in to the web interface and accept the **Terms of Use**. Otherwise, an error will be 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 *ktl* 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 an event for which KUMA has information available, instead of the **Threat Lookup enrichment** window, the data from Kaspersky Threat Intelligence Portal is displayed, with the time when it was received indicated. 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.

Integration with R-Vision Security Orchestration, Automation and Response

R-Vision Security Orchestration, Automation and Response (hereinafter referred to as R-Vision SOAR) is a software platform used for automation of monitoring, processing, and responding to information security incidents. It aggregates cyberthreat data from various sources into a single database for further analysis and investigation to facilitate incident response capabilities.

R-Vision SOAR can be integrated with KUMA. When this integration is enabled, the creation of a KUMA alert triggers the creation of an incident in R-Vision SOAR. A KUMA alert and its R-Vision SOAR incident are interdependent. When the status of an incident in R-Vision SOAR is updated, the status of the corresponding KUMA alert is also changed.

Integration of R-Vision SOAR and KUMA is configured in both solutions. In KUMA integration settings are available only for general administrators.

Mapping KUMA alert fields to R-Vision SOAR incident fields when transferring data via API

KUMA alert field	R-Vision SOAR incident field
`FirstSeen`	`detection`
`priority`	`level`
`correlationRuleName`	`description`
`events` As a JSON file	`files`

Configuring integration in KUMA

This section describes integration of KUMA with R-Vision SOAR from the KUMA side.

Integration in KUMA is configured in the web interface under **Settings** → **Integrations** → **IRP/SOAR**.

To configure integration with R-Vision SOAR:

1. Open the **Resources** section of the KUMA web interface and under **Resources**, select **Secrets**.

   The list of available secrets will be displayed.

2. Click the **Add** button to create a new secret. This resource is used to store token for R-Vision SOAR API requests.

   The secret window is displayed.

3. Enter information about the secret:

   a. In the **Name** field, enter 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 the created resource.

   c. In the **Type** drop-down list, select **token**.

   d. In the **Token** field, enter your R-Vision SOAR API token.

      You can obtain the token in the R-Vision SOAR web interface under **Settings** → **Common** → **API**.

   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 secret.

4. Click **Create**.

   The R-Vision SOAR API token is now saved and can be used in other KUMA resources.

5. In the KUMA web interface, go to the **Settings** → **Integrations** → **IRP / SOAR** section.

   The window containing R-Vision SOAR integration settings opens.

6. Use the **State** toggle switch to enable the integration of the R-Vision SOAR with KUMA. The integration is disabled by default.

7. In the **Secret** drop-down list, select the previously created secret.

   You can create a new secret by selecting **Create**. The created secret is saved in the **Resources** → **Resources** → **Secrets**.

8. In the **URL** specify the URL of the R-Vision SOAR server host.

9. In the **Field name where KUMA alert IDs must be placed**, specify the name of the R-Vision SOAR field where the ID of the KUMA alert must be written.

10. In the **Field name where KUMA alert URLs must be placed** field, specify the name of the R-Vision SOAR field where the link to the KUMA alert must be written.

11. In the **Category** field, specify the category of the R-Vision SOAR incident that is created after alert information is received from KUMA.

12. In the **KUMA event fields that must be sent to IRP / SOAR** drop-down list, select the KUMA event fields to be sent to R-Vision SOAR.

13. Under **Severity**, specify the thresholds for mapping KUMA severity levels to R-Vision SOAR severity levels.

14. Click **Save**.

In KUMA integration with R-Vision SOAR is now configured. If integration is also configured in R-Vision SOAR, when alerts appear in KUMA, information about those alerts will be sent to R-Vision SOAR to create an incident. The **Details on alert** section in the KUMA web interface displays a link to R-Vision SOAR.

If you are working with multiple tenants and want to integrate with R-Vision SOAR, the names of tenants must match the abbreviated names of companies in R-Vision SOAR.

Configuring integration in R-Vision SOAR

This section describes KUMA integration with R-Vision SOAR from the R-Vision SOAR side.

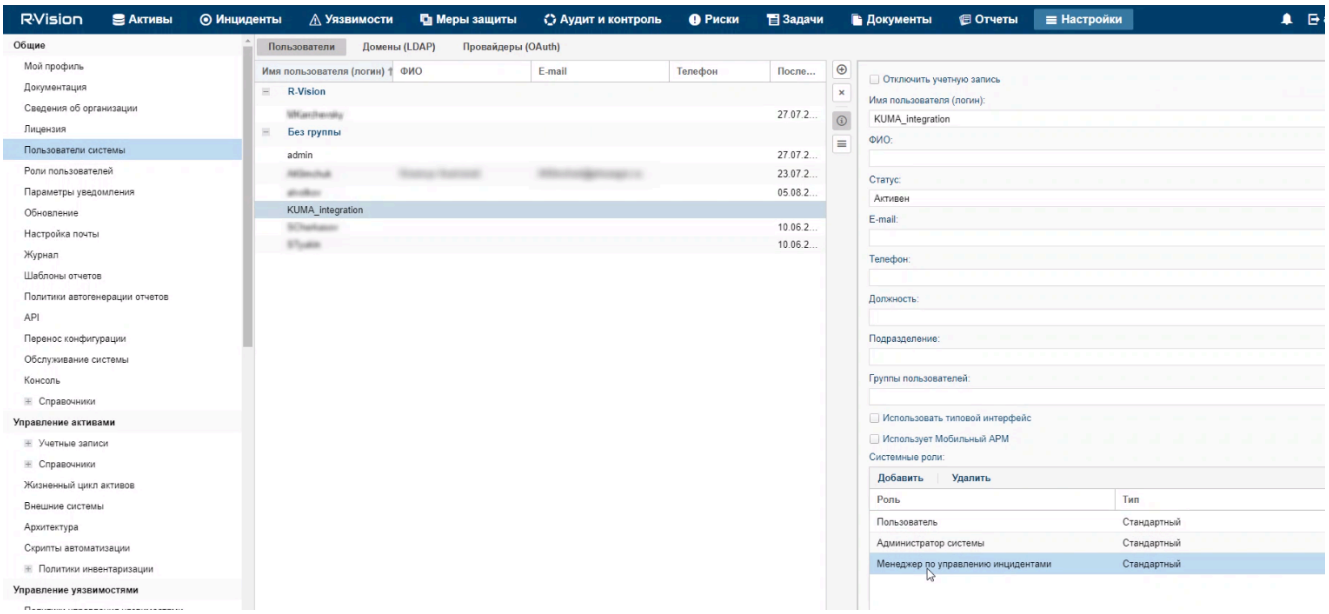
Integration in R-Vision SOAR is configured in the **Settings** section of the R-Vision SOAR web interface. For details on configuring R-Vision SOAR, please refer to the documentation on this application.

Configuring integration with KUMA consists of the following steps:

- **Configuring R-Vision SOAR user role**

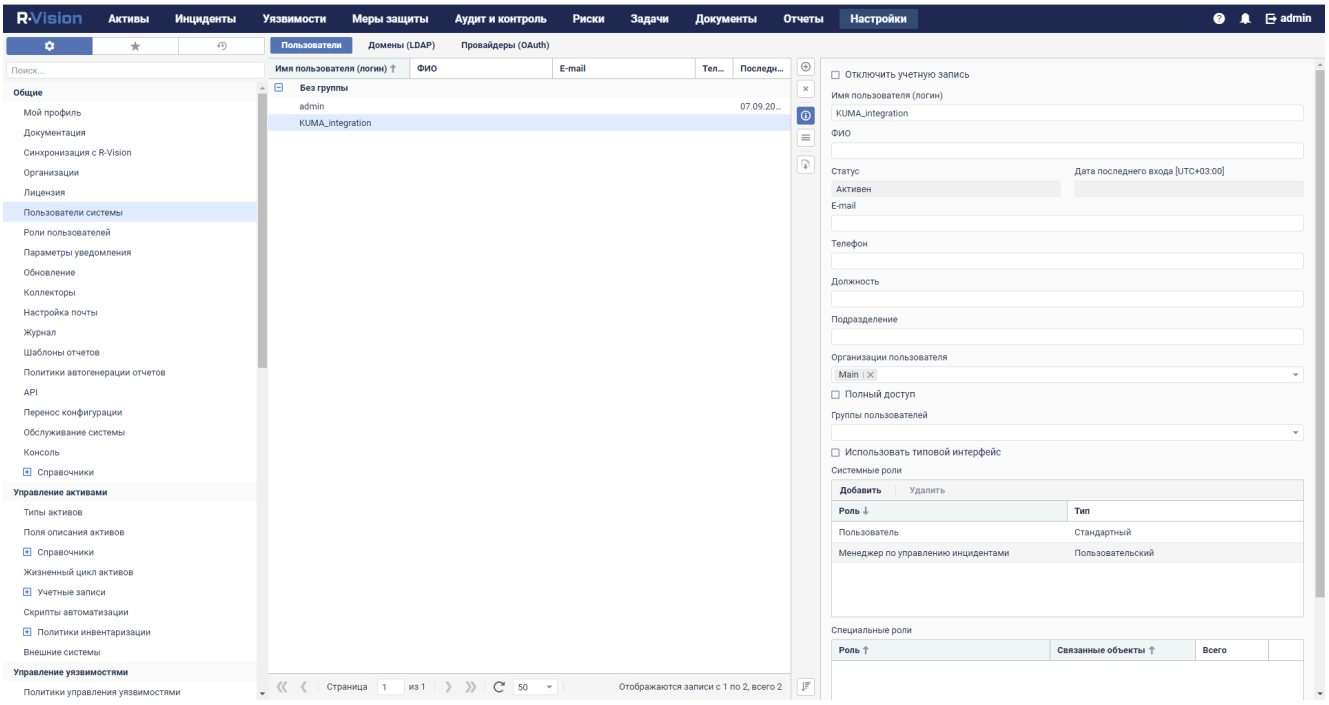
  1. Assign the **Incident manager** system role to the R-Vision SOAR user utilized for integration. The role is assigned when a user is selected in the R-Vision SOAR web interface in the **Settings** → **Common** → **System users** section. The role is added under **System roles** (click the links below for images).

     R-Vision SOAR version 4.0 user with the Incident Manager role



Assigning a role to a user in R-Vision SOAR 4.0

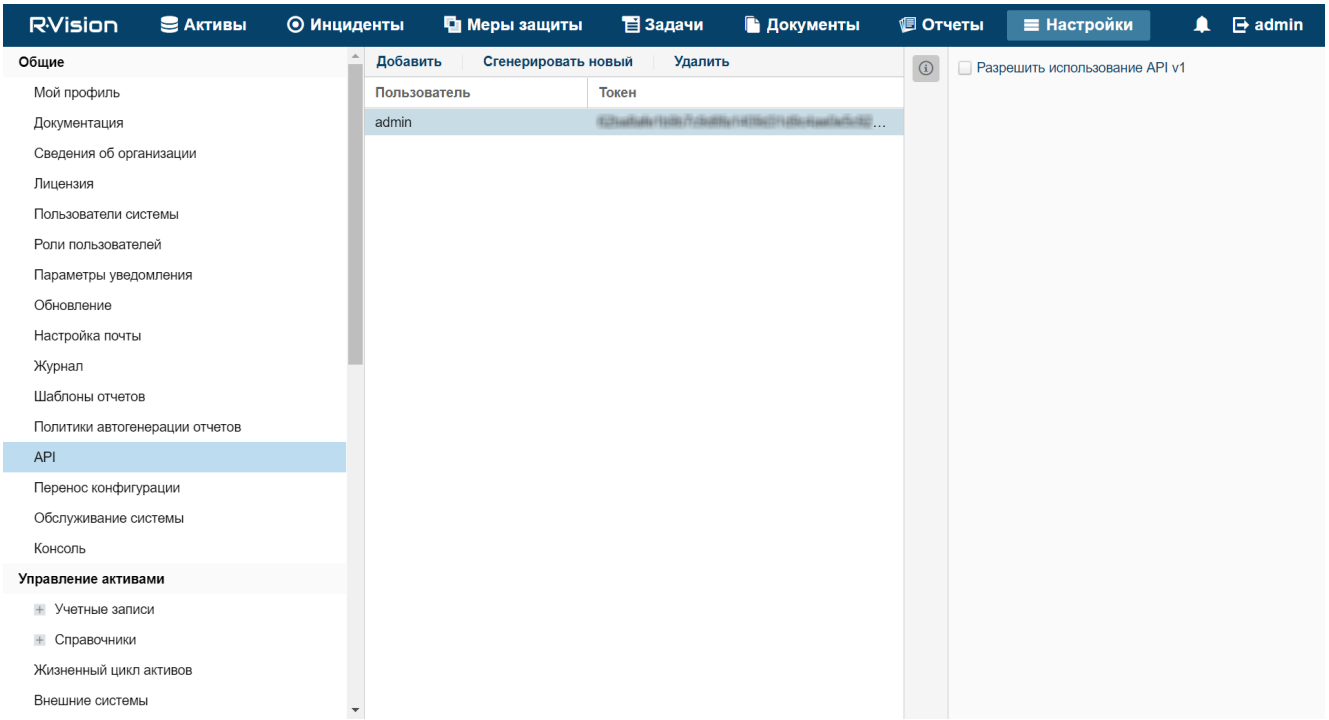
R-Vision SOAR version 5.0 user with the Incident Manager role



Assigning a role to a user in R-Vision SOAR 5.0

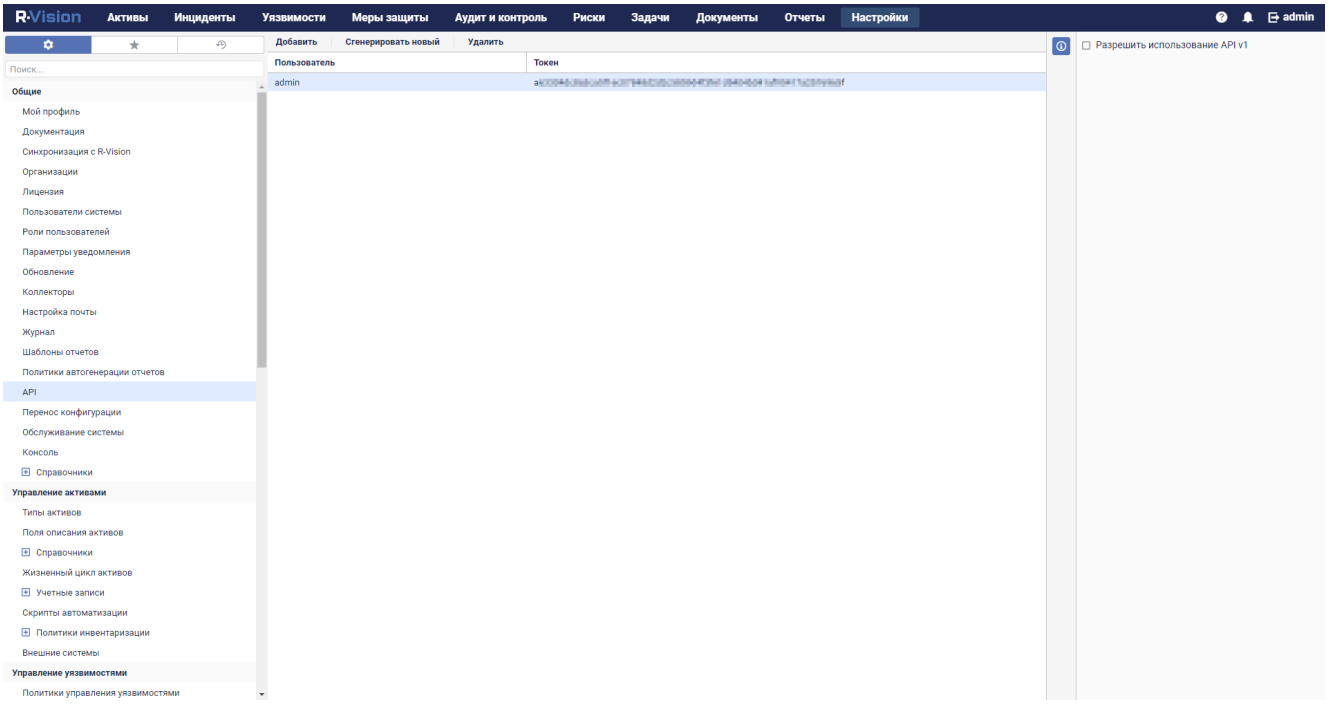
2. Make sure that the API token of the R-Vision SOAR user utilized for integration is indicated in the secret in the KUMA web interface. The token is displayed in the R-Vision SOAR web interface under **Settings** → **Common** → **API** (click the links below for images).

API token in R-Vision SOAR version 4.0



API token in R-Vision SOAR version 4.0

API token in R-Vision SOAR version 5.0



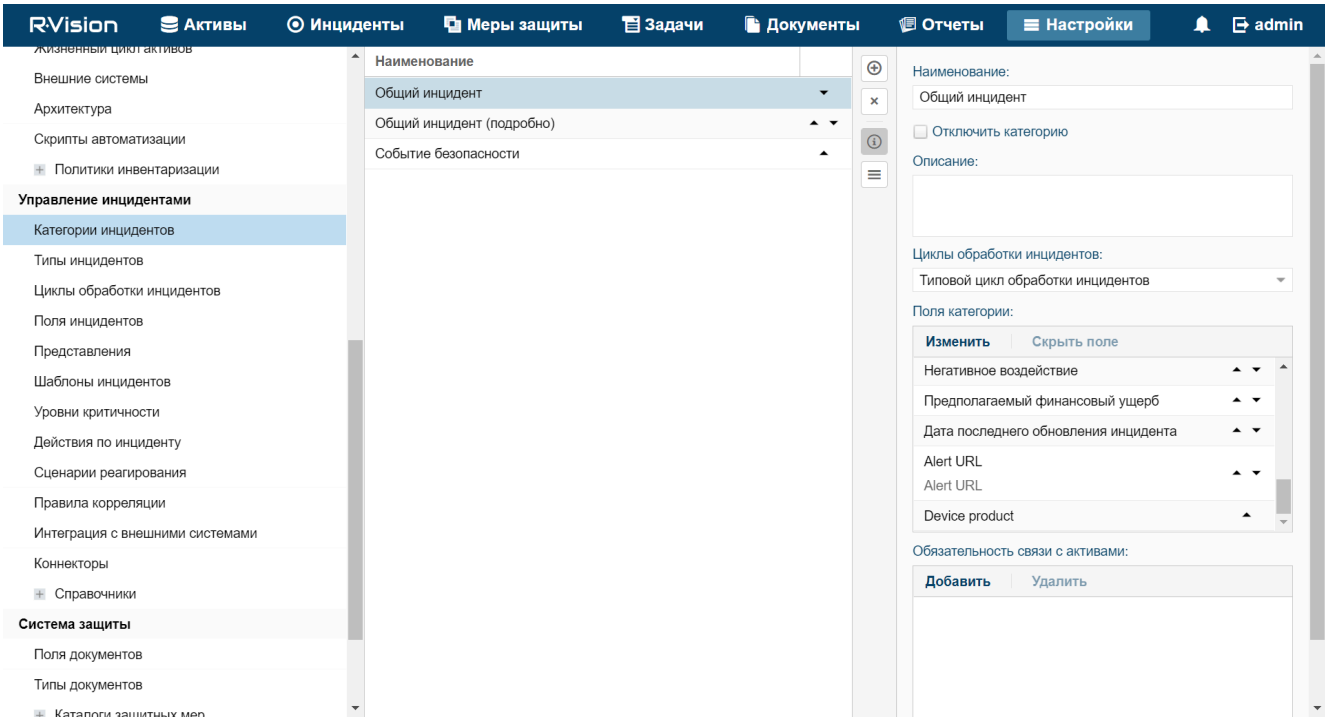
API token in R-Vision SOAR version 5.0

- **Configuring R-Vision SOAR incident fields and KUMA alert fields**

  1. Add the ALERT_ID and ALERT_URL incident fields.

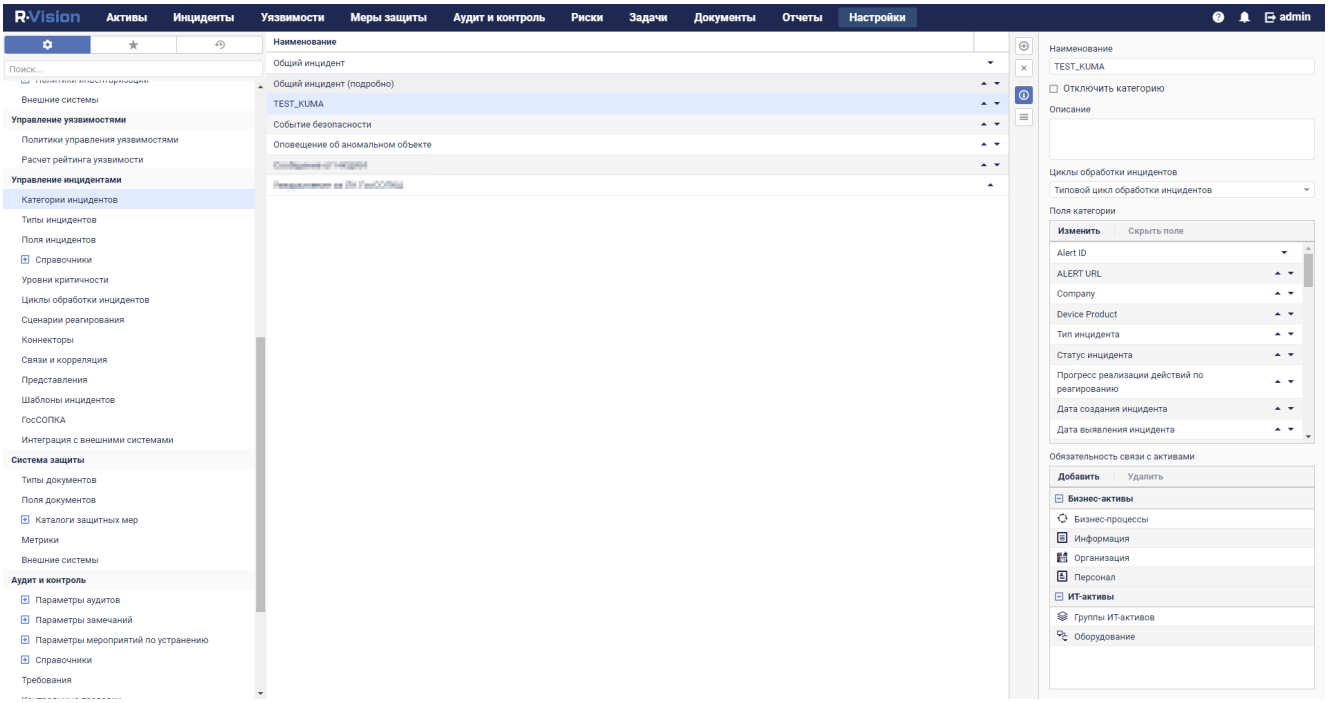
  2. Configure the category of R-Vision SOAR incidents created based on KUMA alerts. You can do this in the R-Vision SOAR web interface, in the **Settings** → **Incident management** → **Incident categories** section. Add a new incident category or edit an existing incident category by indicating the previously created `Alert ID` and `Alert URL` incident fields under **Category fields** (click the links below for images). The `Alert ID` field can be hidden.

     Incident categories with data from KUMA alerts in R-Vision SOAR version 4.0



Configuring fields in incidents in R-Vision SOAR 4.0

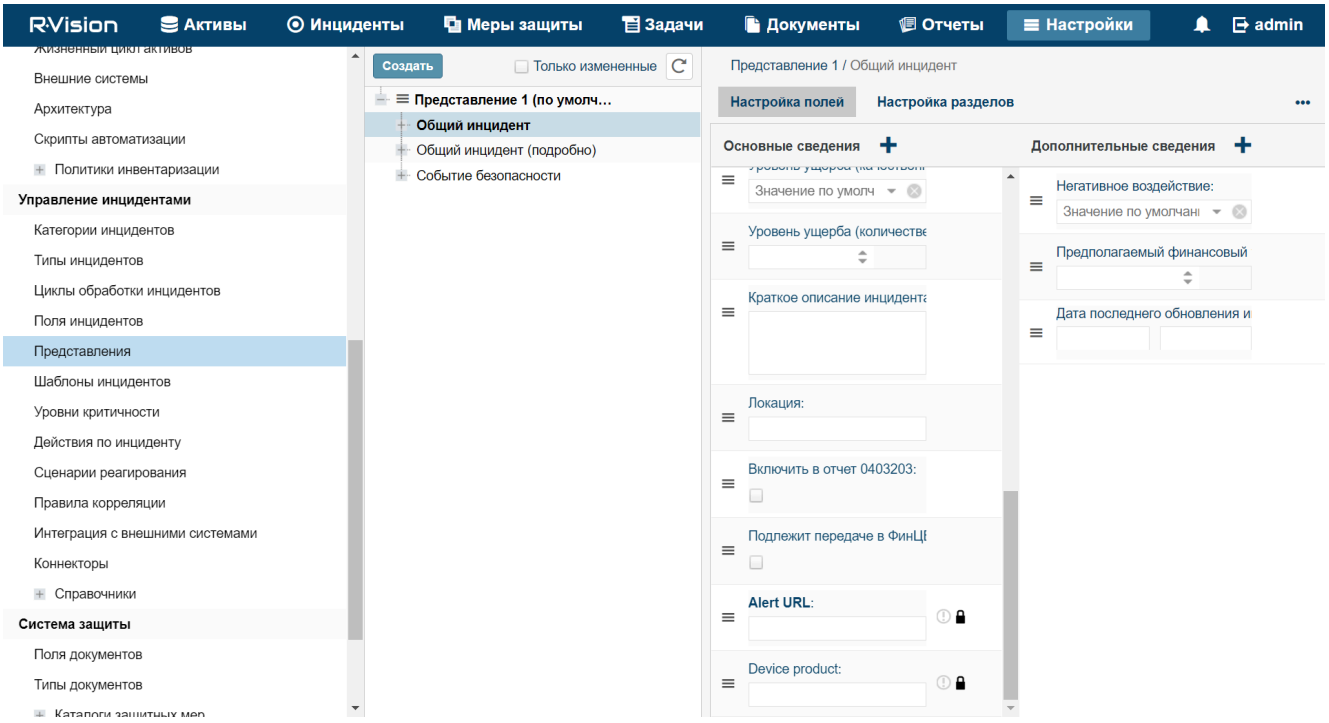
Incident categories with data from KUMA alerts in R-Vision SOAR version 5.0



Configuring fields in incidents in R-Vision SOAR 5.0

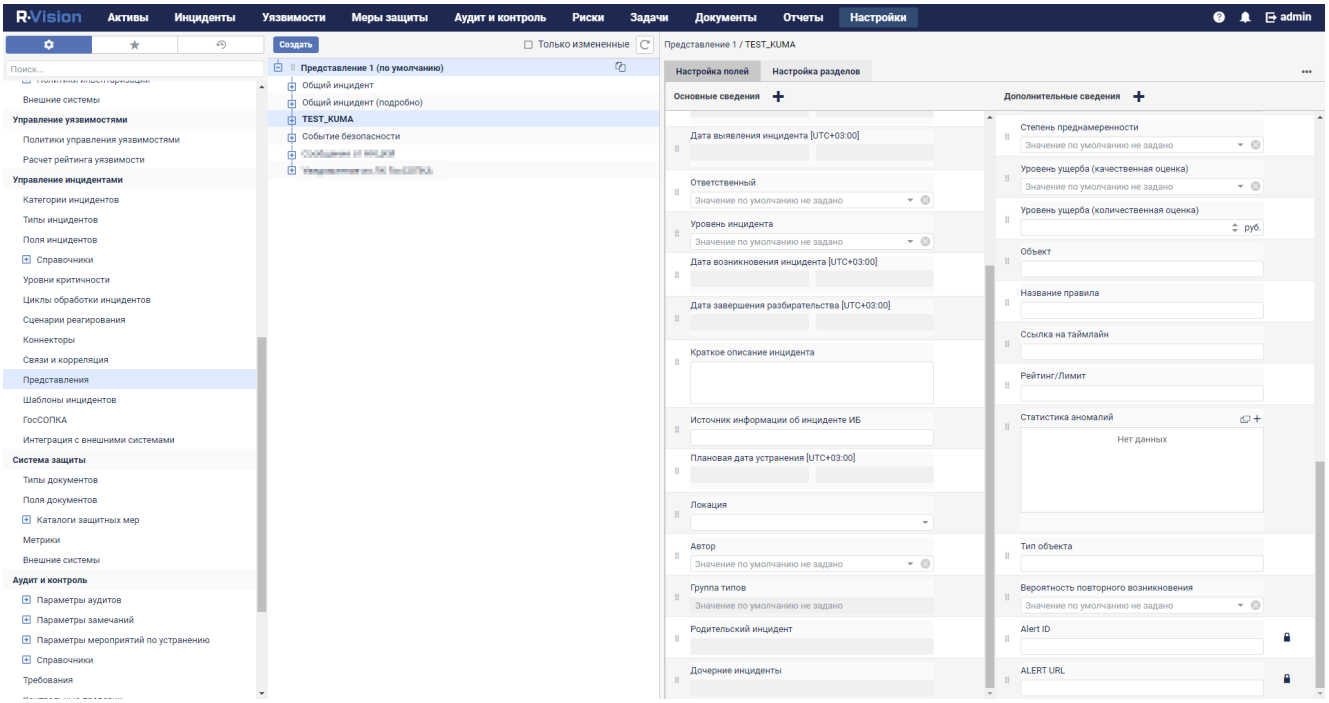
3. Block editing of previously created `Alert ID` and `Alert URL` incident fields. In the R-Vision SOAR web interface, under **Settings** → **Incident management** → **Presentation**, select the category of R-Vision SOAR incidents that will be created based on KUMA alerts and put a lock icon next to the `Alert ID` and `Alert URL` incident fields (click the links below for images).

   The Alert URL field is not editable in R-Vision SOAR version 4.0



Preventing the editing of fields in R-Vision SOAR 4.0

The Alert URL field is not editable in R-Vision SOAR version 5.0



Preventing the editing of fields in R-Vision SOAR 5.0

- **Creating R-Vision SOAR collector and connector**

  1. Create an R-Vision SOAR collector to interact with KUMA.
  2. Create and configure an R-Vision SOAR connector to send API requests to close KUMA alerts.

- **Creating a rule to close a KUMA alert**

  Create a rule for sending KUMA alert closing request when R-Vision SOAR incident is closed.

Integration with KUMA is now configured in R-Vision SOAR. If integration is also configured in KUMA, when alerts appear in KUMA, information about those alerts is sent to R-Vision SOAR to create an incident. The **Details on alert** section in the KUMA web interface displays a link to R-Vision SOAR.

Adding the ALERT_ID and ALERT_URL incident fields

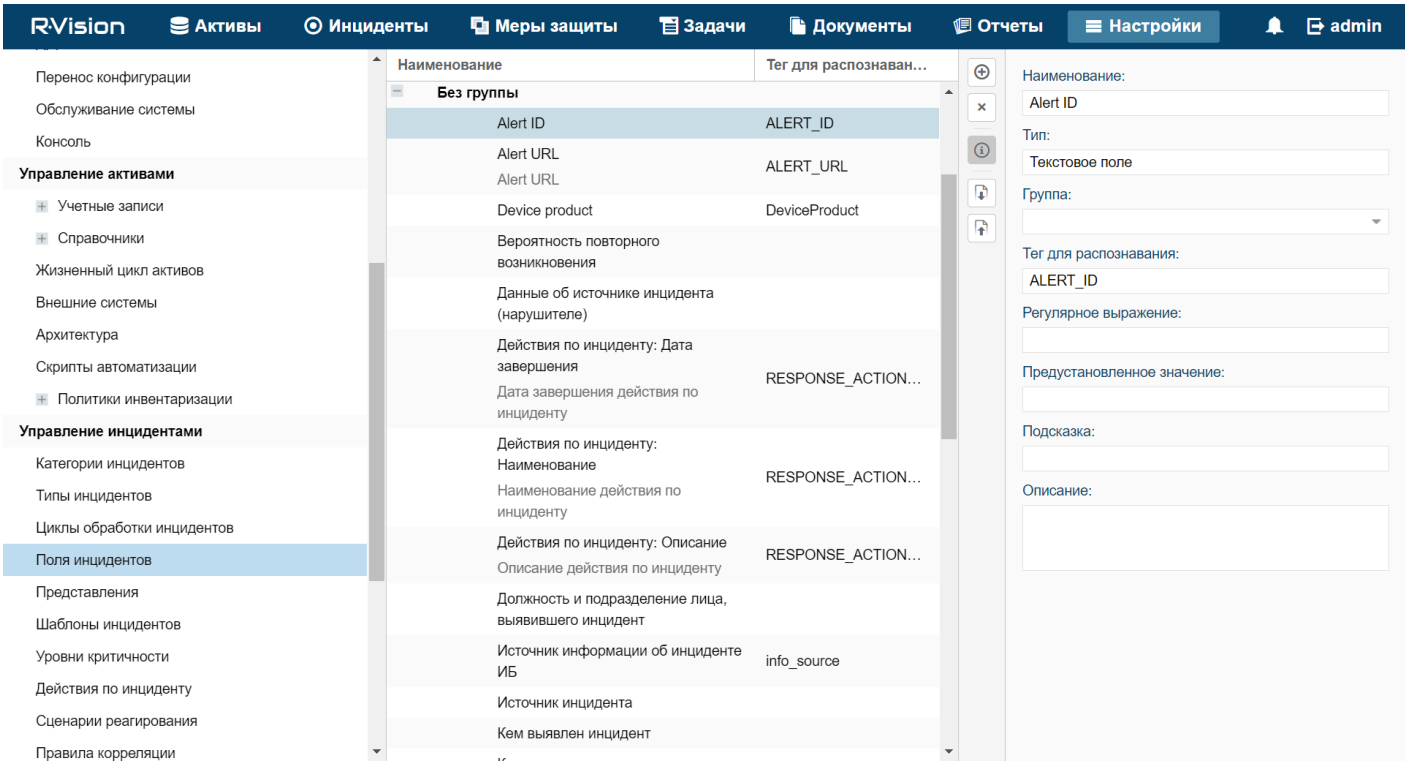
To add the ALERT_ID incident field in the R-Vision SOAR:

1. In the R-Vision SOAR web interface, under **Settings** → **Incident management** → **Incident fields**, select the **No group** group of fields.
2. Click the plus icon in the right part of the screen.

   The right part of the screen will display the settings area for the incident field you are creating.
3. In the **Title** field, enter the name of the field (for example: `Alert ID`).
4. In the **Type** drop-down list, select **Text field**.
5. In the **Parsing Tag** field, enter `ALERT_ID`.

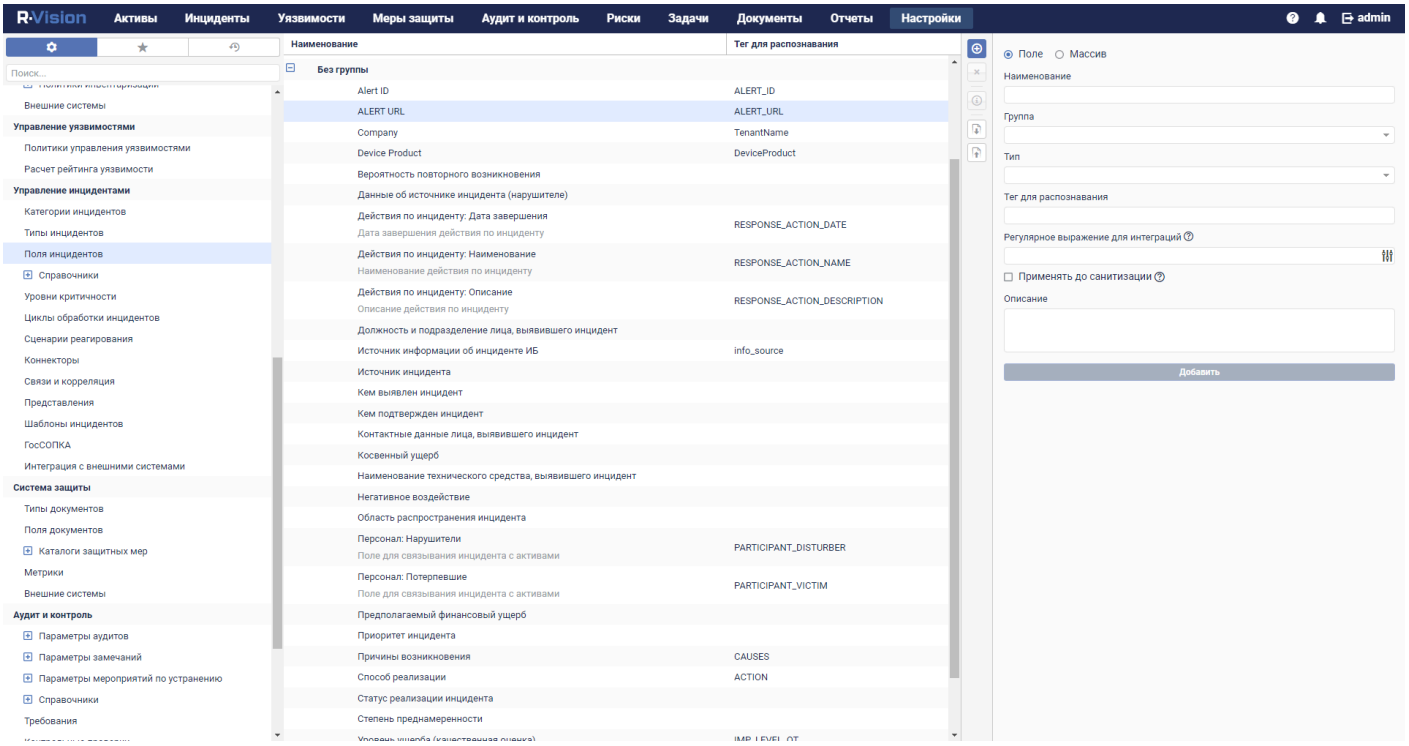
The ALERT_ID field has been added to the R-Vision SOAR incident (click the links below to see images).

ALERT_ID field in R-Vision SOAR version 4.0



Adding the ALERT_ID field in R-Vision SOAR 4.0

ALERT_ID field in R-Vision SOAR version 5.0



Adding the ALERT_ID field in R-Vision SOAR 5.0

To add the ALERT_URL incident field in R-Vision SOAR:

1. In the R-Vision SOAR web interface, under **Settings** → **Incident management** → **Incident fields**, select the **No group** group of fields.

2. Click the plus icon in the right part of the screen.

   The right part of the screen will display the settings area for the incident field you are creating.

3. In the **Title** field, enter the name of the field (for example: `Alert URL`).

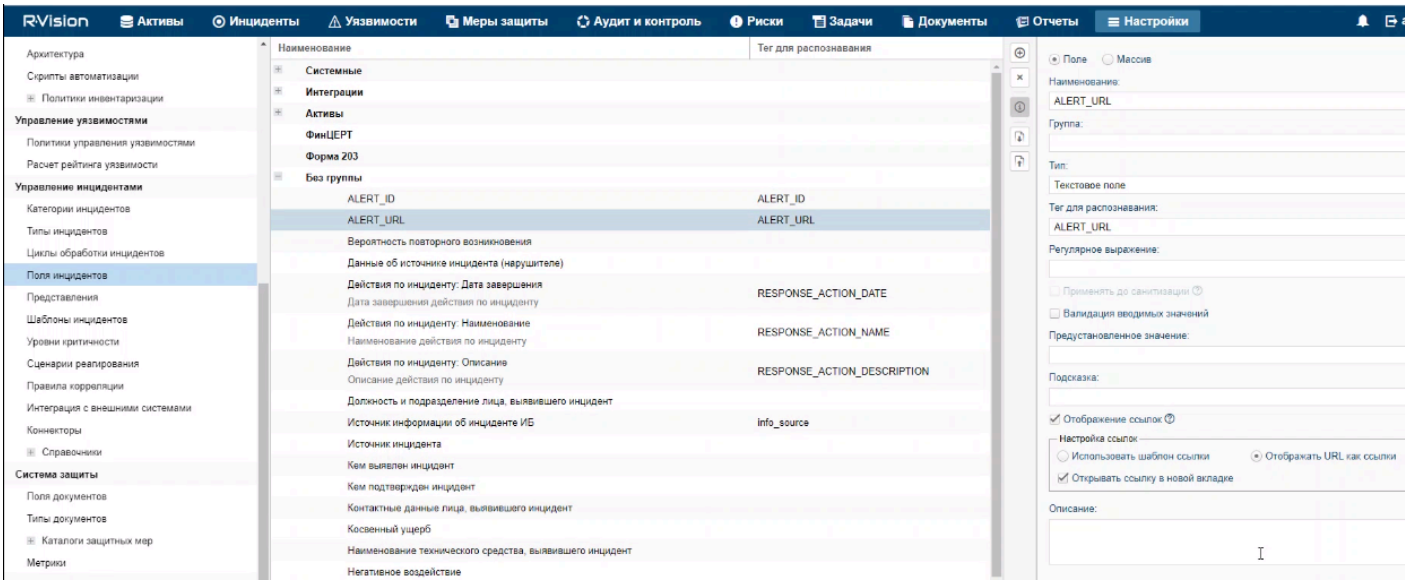
4. In the **Type** drop-down list, select **Text field**.

5. In the **Parsing Tag** field, enter `ALERT_URL`.

6. Select the **Display links** and **Display URL as links** check boxes.

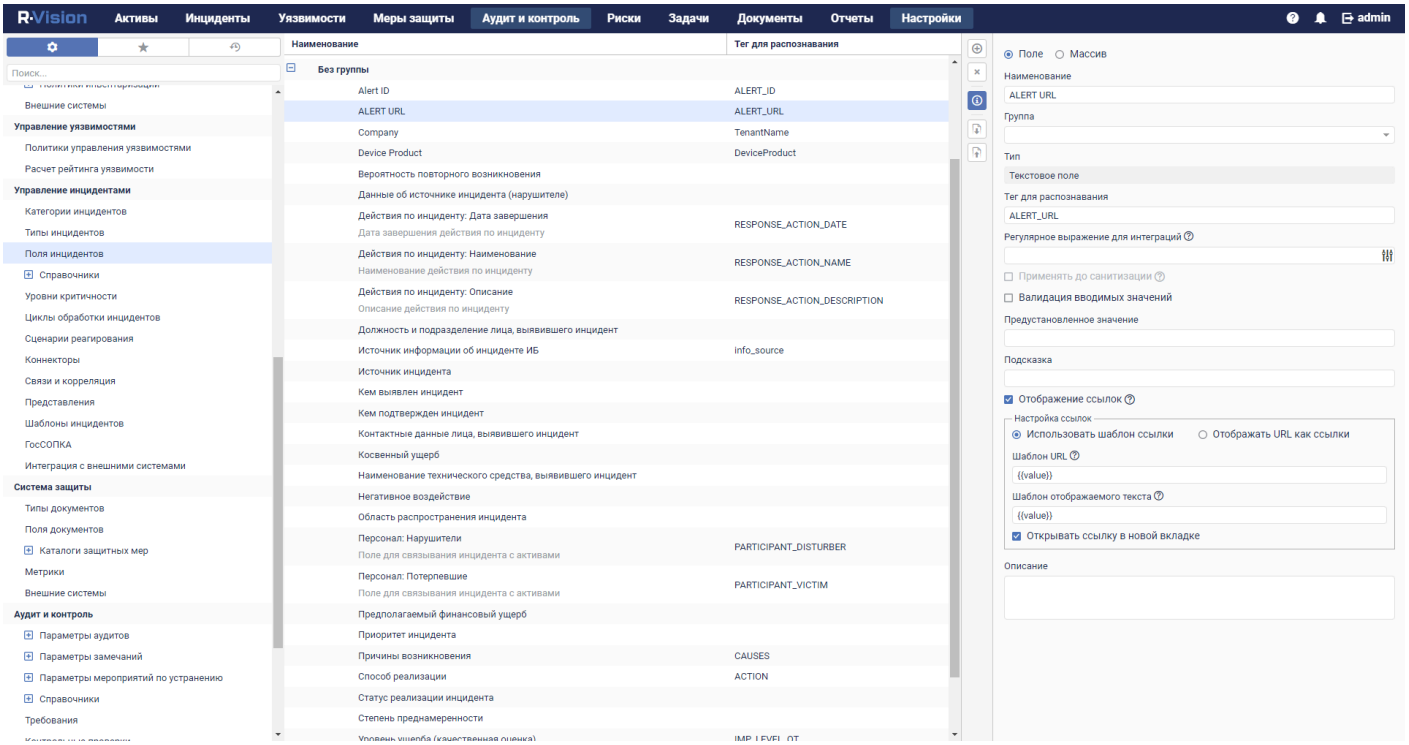
The ALERT_URL field has been added to the R-Vision SOAR incident (click the links below to see images).

ALERT_URL field in R-Vision SOAR version 4.0



Adding the ALERT_URL field in R-Vision SOAR 4.0

ALERT_URL field in R-Vision SOAR version 5.0



Adding the ALERT_URL field in R-Vision SOAR 5.0

If necessary, you can likewise configure the display of other data from a KUMA alert in an R-Vision SOAR incident.

Creating a collector in R-Vision SOAR

To create a collector in R-Vision SOAR:

1. In the R-Vision SOAR web interface, under **Settings** → **Common** → **Collectors**, click the plus icon.
2. Specify the collector name in the **Name** field (for example, `Main collector`).
3. In the **Collector address** field, enter the IP address or hostname where the R-Vision SOAR is installed (for example, `127.0.0.1`).
4. In the **Port** field type `3001`.
5. Click **Add**.
6. On the **Organizations** tab, select the organization for which you want to add integration with KUMA and select the **Default collector** and **Response collector** check boxes.

The R-Vision SOAR collector is created.

Creating connector in R-Vision SOAR

To create connector in R-Vision SOAR:

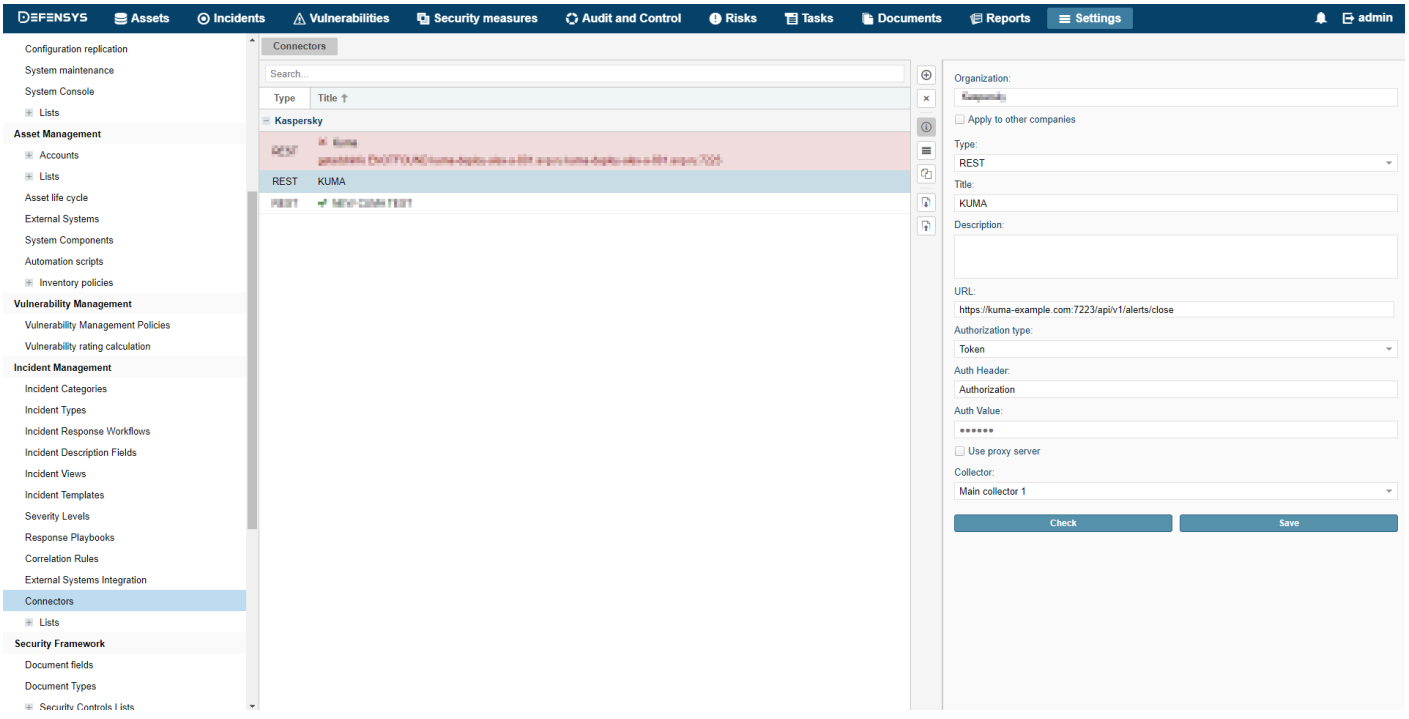
1. In the R-Vision SOAR web interface, under **Settings** → **Incident management** → **Connectors**, click the plus icon.
2. In the **Type** drop-down list, select **REST**.
3. In the **Name** field, specify the connector name, such as `KUMA`.
4. In the **URL** field, enter an API request to close an alert in the following format: `<FQDN of the KUMA Core server>:<port used for API requests (7223 by default)>/api/v1/alerts/close`.

   Example: `https://kuma.example.com:7223/api/v1/alerts/close`
5. In the **Authorization type** drop-down list, select **Token**.
6. In the **Auth header** field type `Authorization`.
7. In the **Auth value** field enter the token of KUMA user with general administrator role in the following format:

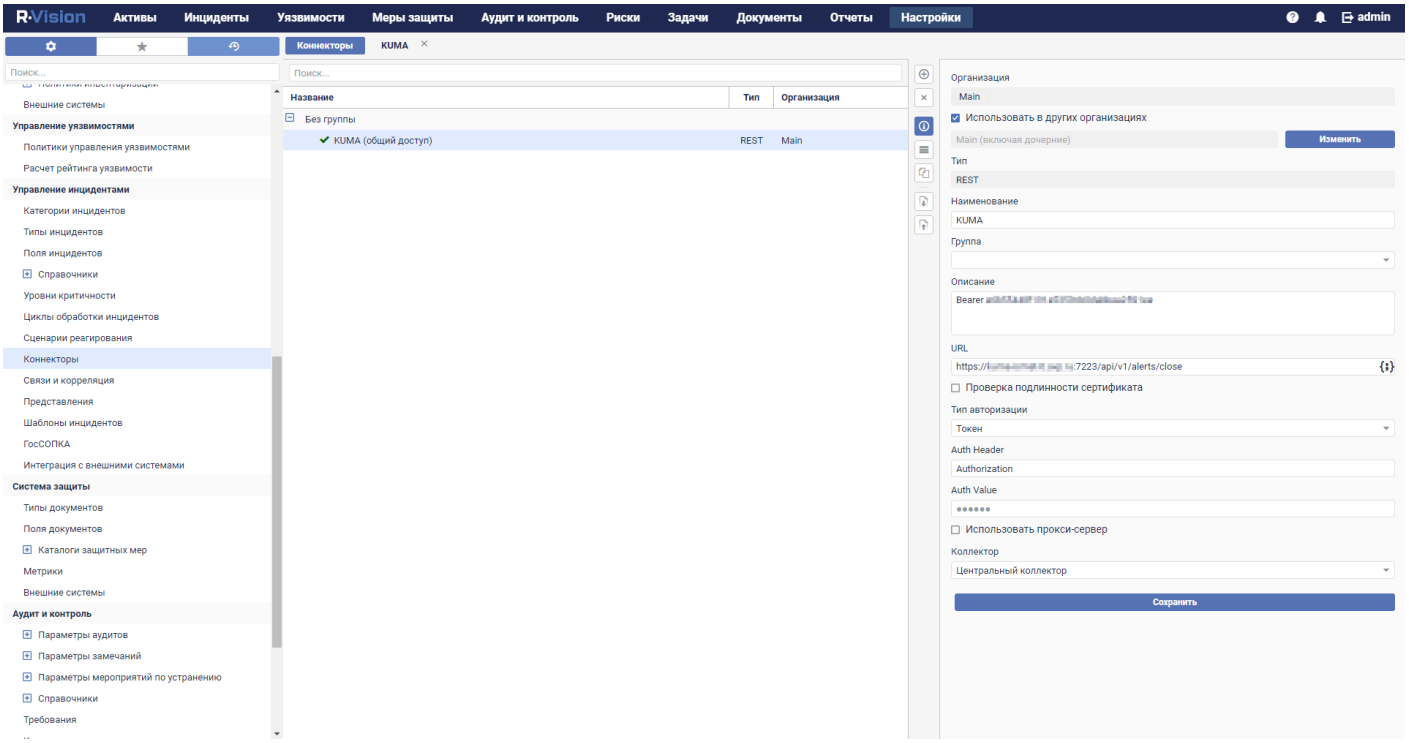
   `Bearer <KUMA General administrator token>`
8. In the **Collector** drop-down list select previously created collector.
9. Click **Save**.

The connector is created (click the links below for images).

Connector in R-Vision SOAR version 4.0



Connector in R-Vision SOAR version 4.0



Connector in R-Vision SOAR version 5.0

When connector is created you must configure sending API queries for closing alerts in KUMA.

To configure API queries in R-Vision SOAR:

1. In the R-Vision SOAR web interface, under **Settings** → **Incident management** → **Connectors**, open for editing the newly created connector.

2. In the request type drop-down list, select **POST**.

3. In the **Params** field, enter the API request to close an alert in the following format: `<FQDN of the KUMA Core server>:<port used for API requests (7223 by default)>/api/v1/alerts/close`.

   For example, `https://kuma.example.com:7223/api/v1/alerts/close`

4. On the **HEADERS** tab, add the following keys and values:

   - Key `Content-Type`; value: `application/json`.
   - Key `Authorization`; value: `Bearer <KUMA general administrator token>`.

     The token of the KUMA general administrator can be obtained in the KUMA web interface under **Settings** → **Users**.

5. On the **BODY** → **Raw** tab, enter the body of the API request:

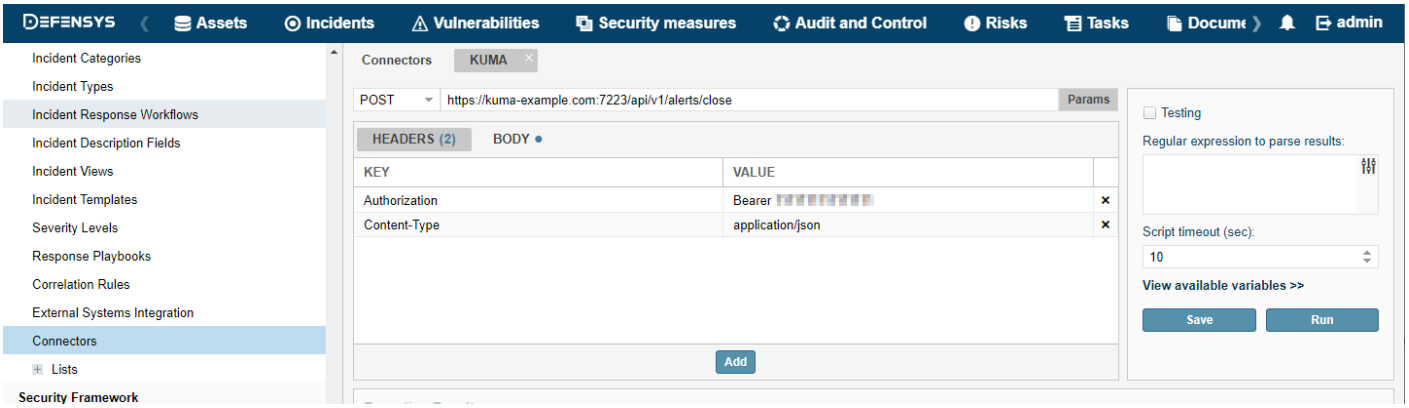
```
{
    "id":"{{tag.ALERT_ID}}",
    "reason":"<Reason for closing the alert. Available values: "Incorrect Correlation Rule", "Incorrect Data", "Responded".> "
```

```
}
```

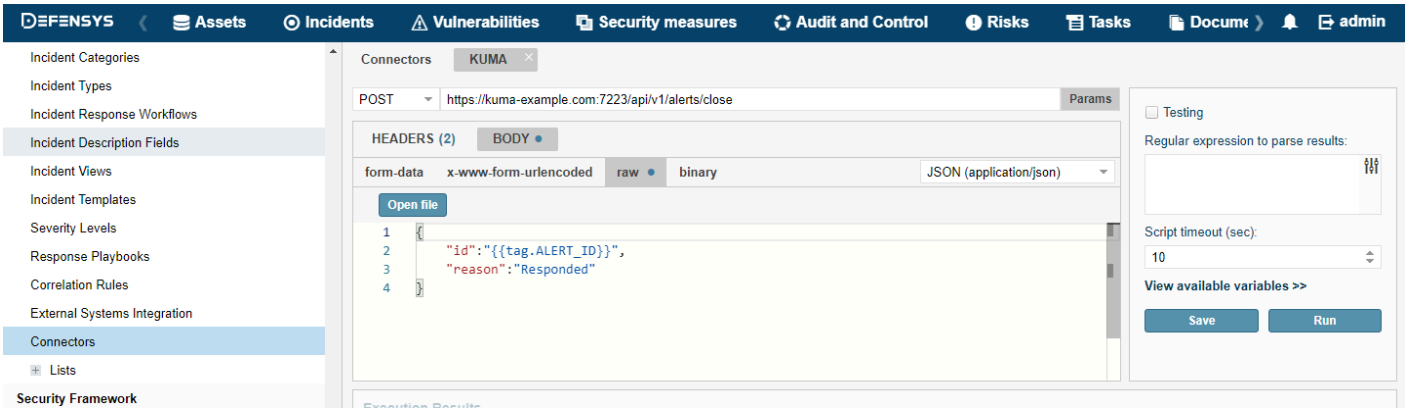
6. Click **Save**.

The connector is configured (click the links below for images).

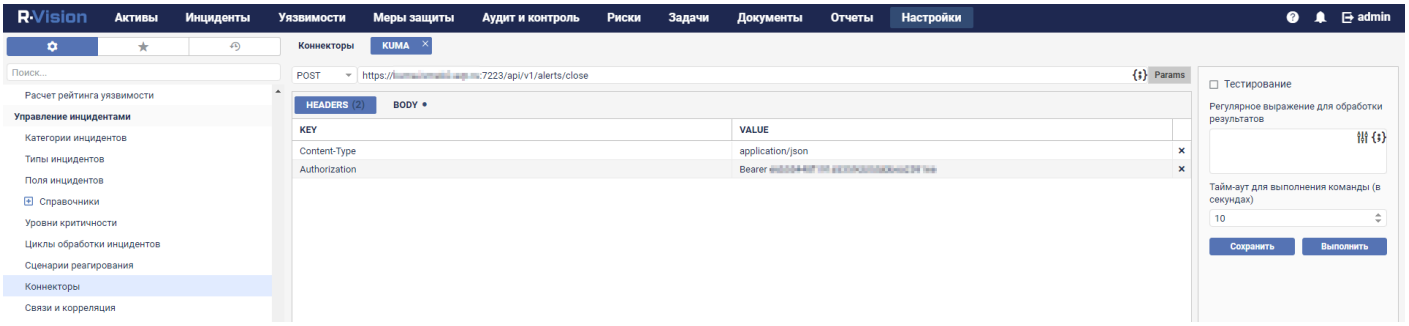
Connector in R-Vision SOAR version 4.0



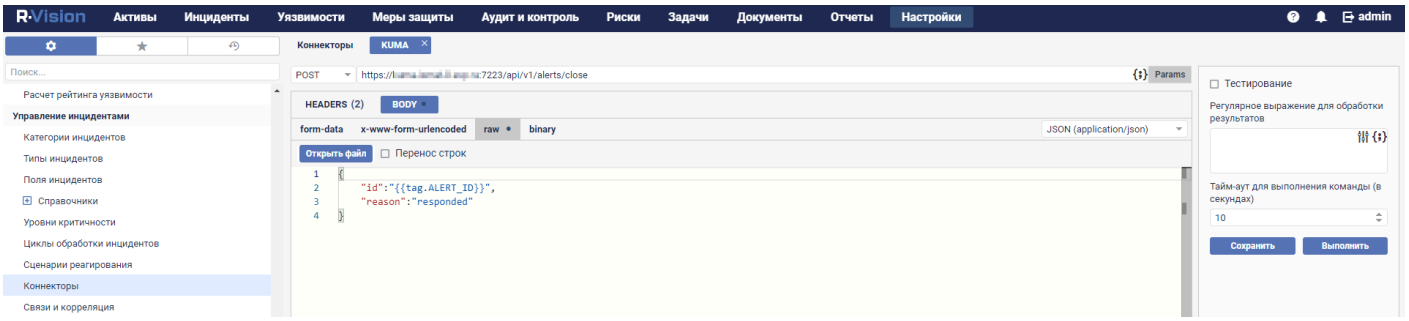
API request header



API request body



API request header



API request body

Creating rule for closing KUMA alert when R-Vision SOAR incident is closed

To create a rule for sending an alert closing request to KUMA when an R-Vision SOAR incident is closed:

1. In the R-Vision SOAR web interface, under **Settings** → **Incident management** → **Response playbooks**, click the plus icon.

2. In the **Name** field, type the name of the rule, for example, `Close alert`.

3. In the **Group** drop-down list select **All playbooks**.

4. Under **Autostart criteria**, click **Add** and enter the conditions for triggering the rule in the opened window:

   a. In the **Type** drop-down list, select **Field value**.

   b. In the **Field** drop-down list, select **Incident status**.

   c. Select the **Closed** status.

   d. Click **Add**.

   Rule trigger conditions are added. The rule will trigger when an incident is closed.

5. Under **Incident Response Actions**, click **Add** → **Run connector**. In the window that is opened, select the connector that you want to be run when the rule is triggered:

   a. In the **Connector** drop-down list select previously created connector.

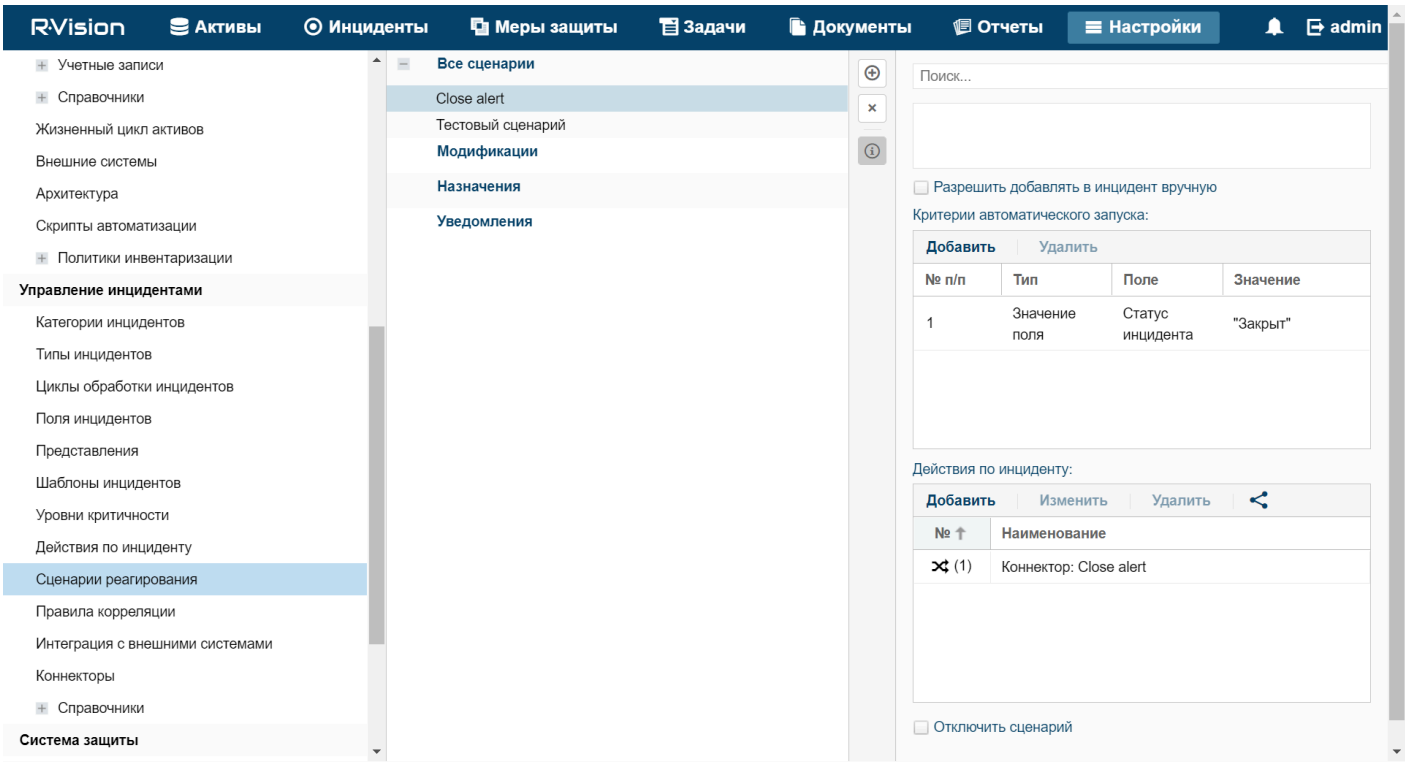
   b. Click **Add**.

Connector added to the rule.

6. Click **Add**.

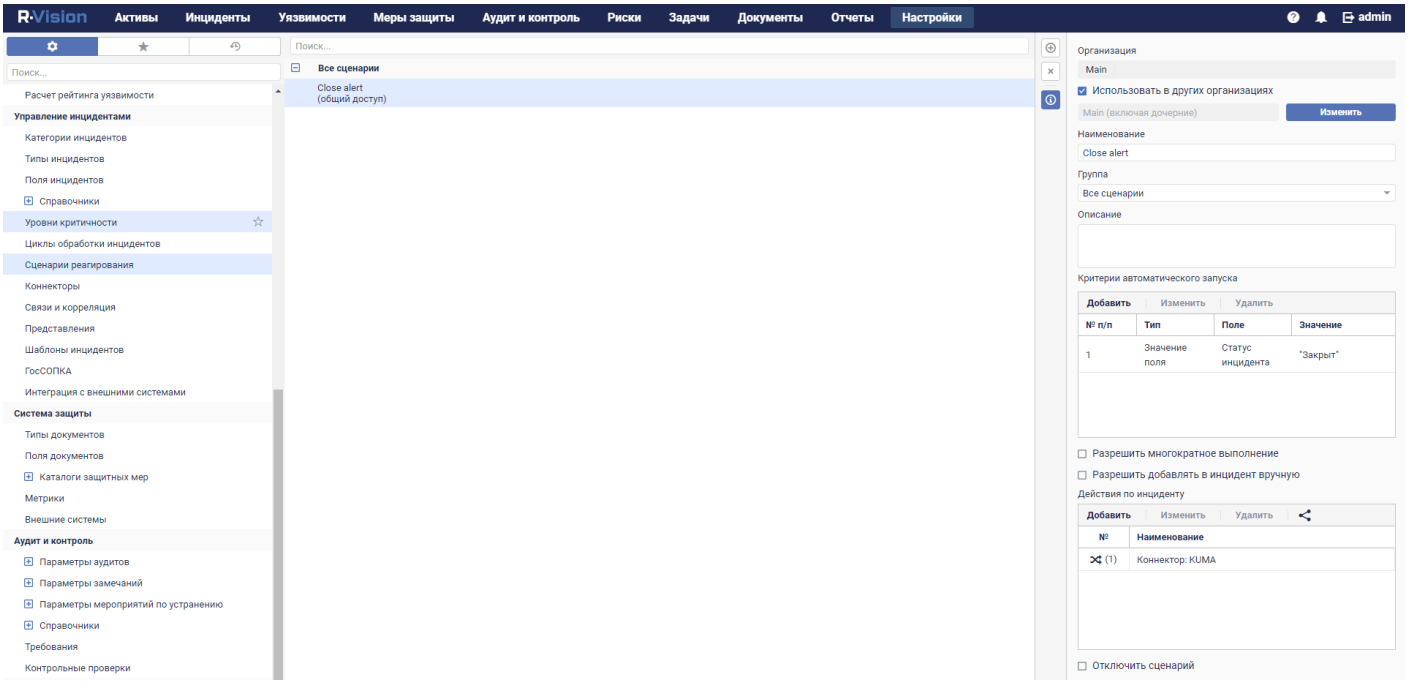
A rule is created for sending a KUMA alert closing request when an R-Vision SOAR incident is closed (click the links below to see images).

R-Vision IRP version 4.0 playbook rule



R-Vision SOAR version 4.0 playbook rule

R-Vision SOAR version 5.0 playbook rule



R-Vision IRP version 5.0 playbook rule

Managing alerts using R-Vision SOAR

After integration of KUMA and R-Vision SOAR is configured, data on KUMA alerts starts coming into R-Vision SOAR. Changes of alert parameters in KUMA are reflected in R-Vision SOAR. Any changes in the statuses of alerts in KUMA or R-Vision SOAR (except closing an alert) are also reflected in the other system.

Alert management scenarios when KUMA and R-Vision SOAR are integrated:

- **Send cyberthreat data from KUMA to R-Vision SOAR**

  Data on detected alerts is automatically sent from KUMA to R-Vision SOAR. An incident is also created in R-Vision SOAR.

  The following information about the KUMA alert is sent to R-Vision SOAR:

  - ID.
  - Name.
  - Status.
  - Date of the first event related to the alert.
  - Date of the last detection related to the alert.
  - User account name or email address of the security officer assigned to process the alert.
  - Alert severity.
  - Category of the R-Vision SOAR incident corresponding to the KUMA alert.
  - Hierarchical list of events related to the alert.
  - List of alert-related assets (internal and external).
  - List of users related to the alert.
  - Alert change log.
  - Link to the alert in KUMA.

- **Investigate cyberthreats in KUMA**

  Initial processing of an alert is performed in KUMA. The security officer can update and change any parameters of an alert except its ID and name. Any changes are reflected in the R-Vision SOAR incident card.

  If a cyberthreat turns out to be a false positive and its alert is closed in KUMA, its corresponding incident in R-Vision SOAR is also automatically closed.

- **Close incident in R-Vision SOAR**

  After all necessary work is completed on an incident and the course of the investigation is recorded in R-Vision SOAR, the incident is closed. The corresponding KUMA alert is also automatically closed.

- **Open a previously closed incident**

  If active monitoring detects that an incident was not completely resolved or if additional information comes up, this incident is re-opened in R-Vision SOAR. However, the alert remains closed in KUMA.

The security officer can use a link to navigate from an R-Vision SOAR incident to the corresponding alert in KUMA and make the necessary changes to any of its parameters except the ID, name, and status of the alert. Any changes are reflected in the R-Vision SOAR incident card.

Further analysis is performed in R-Vision SOAR. When the investigation is complete and the incident is closed again in R-Vision SOAR, the status of the corresponding alert in KUMA remains closed.

- **Request additional data from the source system as part of the response playbook or manually**

  If additional information is required from KUMA when analyzing incidents in R-Vision SOAR, in R-Vision SOAR, you can create a search request to KUMA (for example, you can request telemetry data, reputation, host information). This request is sent via KUMA REST API and the response is recorded in the R-Vision SOAR incident card for further analysis and reporting.

  This same sequence of actions is performed during automatic processing if it is not possible to immediately save all information on an incident during an import.

Integration with Active Directory, Active Directory Federation Services and FreeIPA

You can integrate KUMA with the Active Directory®, Active Directory Federation Services, and FreeIPA services used in your organization.

You can configure a connection to the Active Directory catalog service over the LDAP protocol. This lets you use information from Active Directory in correlation rules for enrichment of events and alerts, and for analytics.

If you configure a connection to a domain controller server, you can use domain authorization. In this case, you can bind the domain groups of users to the KUMA role filters. The users belonging to these groups will be able to use their domain account credentials to log in to the KUMA web interface and will obtain access to solution sections based on their assigned role.

> It is recommended to create the groups of users in Actions Active Directory, Active Directory Federation Services, or FreeIPA in advance if you want to provide such groups with the capability for authorization using their domain account in the KUMA web interface. An email address must be indicated in the properties of a user account in Active Directory.

Connecting over LDAP

LDAP connections are created and managed in the **Settings** → **Integrations** → **LDAP server** section of the KUMA web interface. 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 KUMA web interface, open the **Settings** → **Integrations** → **LDAP server** section and select the tenant for which you want to enable or disable all LDAP connections.
2. Do one of the following:
   a. Set the toggle switch in the **Status** column to **Enabled** or **Disabled**.
   b. Click the name of the tenant in the **Tenant** column to open the **LDAP server integration** sidebar, and in this sidebar:
      - Set the **Status** toggle switch to **Enabled** or **Disabled**.
      - Click **Save**.

To enable or disable a specific LDAP connection:

1. In the KUMA web interface, open the **Settings** → **Integrations** → **LDAP server** section and select the tenant for which you want to enable or disable all LDAP connections.
2. Click the name of the tenant in the **Tenant** column to open the **LDAP server integration** sidebar, and in this sidebar, go to the **Connections** section.
3. Use the toggle switch in the **Status** column to enable or disable a specific LDAP 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 KUMA web interface, open the **Settings** → **Integrations** → **LDAP server** section.

   The **LDAP server integration by tenant** window is displayed.

2. Click the **Add settings for a new tenant** button.

   The **LDAP server integration** panel will open on the right.

3. In the **Tenant** drop-down list, select the tenant that you need to add.

4. Click **Add**.

The selected tenant will be added to the LDAP server integration tenant list.

To remove a tenant from the list of tenants for integration with an LDAP server:

1. In the KUMA web interface, open the **Settings** → **Integrations** → **LDAP server** section.

   The **LDAP server integration by tenant** window is displayed.

2. Select the check box next to the tenant that you want to remove, and click **Remove**.

3. Confirm removal of the tenant.

The selected tenant will be removed from the LDAP server integration tenant list.

Creating an LDAP server connection

To create a new LDAP connection to Active Directory:

1. In the KUMA web interface, open the **Settings** → **Integrations** → **LDAP server** section.

2. Select or add a tenant for which you want to create an LDAP connection.

3. In the **LDAP server integration by tenant** pane, select a tenant from the list or add a new tenant by clicking **Add settings for a new tenant** for which you want to create an LDAP connection.

4. Under **Connections**, click the **Create** button.

   This opens the **Create connections** panel.

5. To use this LDAP connection, turn on the connection using the **State** toggle switch.

6. Specify a unique name for the LDAP connection. The length of the string must be 1 to 128 Unicode characters.

7. In the drop-down list, select a secret containing the credentials for connecting to the Active Directory server. To do so, follow one of these steps:

   - If you previously created a secret, in the **Secret** drop-down list, select the existing secret of the **credentials** type.

   - If you want to create a secret, click **Create**.

     This opens the **Create secret** panel.

     1. In the **Name** field, enter the name of the secret. It must contain 1 to 128 Unicode characters.

     2. In the **User** and **Password** fields, enter the 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>`.

3. If necessary, select the tags for the secret from the **Tags** drop-down list.

4. If necessary, in the **Description** field, enter up to 4,000 Unicode characters describing the secret.

5. Click the **Create** button.

8. In the **URL** field, enter the address of the domain controller in the `<hostname or IP address of server>:<port>` format.

   If necessary, click the **Add** button to specify the addresses of several domain controller servers that may be needed if one of the servers is unavailable. All of the specified servers must reside in the same domain. Only one domain controller server can be specified in one field. If the first domain controller is unavailable, the system tries to connect to the next domain controller in the list, then the next again, until a connection can be established.

   If the host of the domain controller is not found when attempting to connect, for example, if a non-existent address is entered in the **URL** field, there is no switchover to the next domain controller address. KUMA returns an error instead.

9. Select one of the following TLS encryption types for connecting to the domain controller:

   - **LDAPS**.

     When using LDAPS, an encrypted connection is immediately established over port 636. This type of TLS encryption is configured by default.

   - **startTLS**.

     When the startTLS method is used, first it establishes an unencrypted connection to port 389, and then 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.

   - **Insecure**.

   When using an encrypted connection, it is impossible to specify an IP address as a URL.

10. In the **Certificate** drop-down list, specify an existing TLS-certificate of an accredited certification authority that was used to sign the LDAP server's certificate. Custom certificates cannot be used.

    To do so, follow one of these steps:

    - If you already have a certificate, select it from the drop-down list.

    - If you want to create a certificate, click **Create**.

      This opens the **Create secret** panel.

      a. In the **Name** field, enter the name to be displayed in the list of certificates.

      b. Click the **Upload certificate** button to add a file with the Active Directory certificate.

      Base64-encoded X.509 certificate public keys are supported.

      c. If necessary, select the tags for the certificate from the **Tags** drop-down list.

      d. If necessary, enter a description of the certificate in the **Description** field.

      e. Click the **Create** button.

The certificate will be uploaded and displayed in the **Certificate** list.

The **Certificate** field is optional if you have selected the **Insecure** TLS encryption type.

11. 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 specified in the **URL** field, KUMA waits for a response from the first server for the specified time in seconds. If no response is received during that time, the solution attempts to connect to the next server. If none of the indicated servers responds during the specified amount of time, the connection will be terminated with an error.

12. In the **Base DN** field, enter the base distinguished name of the directory in which you need to run the search query.

13. In the **Custom AD account attributes** field, specify additional attributes to be used to enrich events.

    How to specify additional 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.

       Default 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.

       List of attributes imported from AD

       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 attributes** drop-down list in the collector will be automatically appended with 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, delete the row containing the attribute from the collector's mapping table manually. 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.

14. Click the **Create** button.

    If the tenant is deleted while creating a connection, the **New** button remains active, but an error message appears when you try to create a connection. The deleted tenant is displayed in the list of available tenants until you refresh the page by pressing the **F5** key.

The LDAP connection to Active Directory will be created and displayed in the **LDAP server integration** panel.

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 application does not check whether the port is unique.

Creating a copy of an LDAP server connection

You can create an LDAP connection for another tenant 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 KUMA web interface, open the **Settings** → **Integrations** → **LDAP server** section and select the tenant for which you want to copy an LDAP connection.
2. Select a connection and click the **Add settings for a new tenant** button.
3. This opens the **LDAP server integration** window; in that window, in the **Tenant** drop-down list, select the tenant for which you want to create a copy of the connection.
4. Click the **Run import task** button.
5. 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 application does not check whether the port is unique.

Changing an LDAP server connection

To change an LDAP server connection:

1. In the KUMA web interface, open the **Settings** → **Integrations** → **LDAP server** section.
2. Select the tenant for which you want to change the LDAP server connection.

   The **LDAP server integration** panel will open on the right.

3. Click the LDAP server connection that you want to change.

   The window with the settings of the selected LDAP server connection opens.

4. Make the necessary changes to the settings.

5. Click the **Save** button.

   The LDAP server connection will be changed.

6. 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 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 web interface.

To change the schedule of KUMA queries to LDAP servers:

1. In the KUMA web interface, open the **Settings** → **Integrations** → **LDAP server** section.
2. In the **LDAP server integration by tenant** window, select a tenant.

   The **LDAP server integration** window opens.

3. In the **Data refresh interval in hours** field, specify the required interval in hours. The default value is 12.

The query schedule will be 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 KUMA web interface, open the **Settings** → **Integrations** → **LDAP server** section.
2. In the **LDAP server integration by tenant** window, select a tenant.

   The **LDAP server integration** window opens.
3. 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 KUMA web interface, open the **Settings** → **Integrations** → **LDAP server** section.
2. In the **LDAP server integration by tenant** window, select a tenant.

   The **LDAP server integration** window opens.
3. Click the **Run import task** button.

A **task** to receive account data from the selected tenant is added to the Task manager section of the KUMA web interface.

To start an account data update task for one LDAP connection of a tenant:

1. In the KUMA web interface, open the **Settings** → **Integrations** → **LDAP server** section.
2. In the **LDAP server integration by tenant** window, select a tenant.

   The **LDAP server integration** window opens.
3. Under **Connections**, select a connection to the LDAP server.
4. This opens a window; in that window, click the **Run import task** button.

A task to receive account data from the selected connection of the tenant is added to the **Task manager** section of the KUMA web interface.

Deleting an LDAP server connection

To delete LDAP connection to Active Directory:

1. In the KUMA web interface, open the **Settings** → **Integrations** → **LDAP server** section.
2. In the **LDAP server integration by tenant** window, select a tenant.
3. Click the **Delete** button.
4. Confirm deletion of the connection.

The LDAP connection to Active Directory will be deleted.

Authentication using domain accounts

To enable users to perform authentication in the KUMA web interface using their own domain account credentials, perform the following configuration steps.

1. **Enable domain authentication**

   Domain authorization is disabled by default, and a domain connection is not configured.

2. **Configure a connection to the domain controller.**

   The following connections are available:

   - Active Directory (AD)
   - Active Directory Federation Services (ADFS)
   - FreeIPA

   The AD and ADFS connection settings can be configured at the same time.

   You can connect to one domain only.

3. **Add groups of user roles.**

   You can specify a domain group for each KUMA role. After performing authentication using their domain accounts, the users from this group get access to the KUMA web interface in accordance with the specified role.

   The solution checks whether the user's group matches the specified filter in the following order of precedence of roles in the KUMA web interface: Junior analyst → Tier 1 analyst → Tier 2 analyst → Tenant administrator → General administrator. Upon the first match, the application assigns a role to the user and does not check any further. If a user matches two groups in the same tenant, the role with the most privileges is used. If multiple groups are matched for different tenants, the user will be assigned the specified role in each tenant.

   When you click on **Tenant**:

   - Under the tenant, the entire list of roles and groups that belong to this tenant is expanded.
   - Under the row that contains the name of the tenant, the total number of roles and the total number of groups belonging to this tenant, rows with the name of the tenant, the name of the role and the name of the group are displayed.

- Duplicate names of roles and groups are merged when copying and adding a filter (that is, if a **junior analyst role, group 1** belongs to a tenant, when trying to add or copy the same entry (**junior analyst role**, **group 1**) from another tenant, only one **junior analyst role**, **group 1** entry will be present in the table, without duplicates.

- Roles are sorted in descending order of seniority.

You can copy configured user roles and their corresponding groups to a different tenant. To do this, select the check box next to these and click the **Copy to clipboard** button. This opens a window with a list of available tenants. When copying user roles, you can:

- Select one or more roles within a tenant to copy settings.

- Copy an entire tenant with its configured roles and groups.

- Copy settings to multiple tenants at the same time, but only from one tenant. The Shared tenant and the tenant from which you are copying cannot be selected in the list of available tenants.

After their first authentication, all domain users gain access to the default space, which is specified in the **Access to spaces** section.

If necessary, you can delete multiple tenants or user roles with their groups at the same time. To do so, select check boxes next to the entities that you want to delete and click the **Delete** button. You can delete selected roles within a tenant or from different tenants.

Special considerations for logging in after configuring domain authentication

For successful authentication, the following conditions must be met:

- **FreeIPA**: when logging into the system, the user must capitalize the domain name in the login. Example: user@FREEIPA.COM.

- **AD/ADFS**: when logging into the system, the user must specify UserPrincipalName in the login. Example: user@domain.ru.

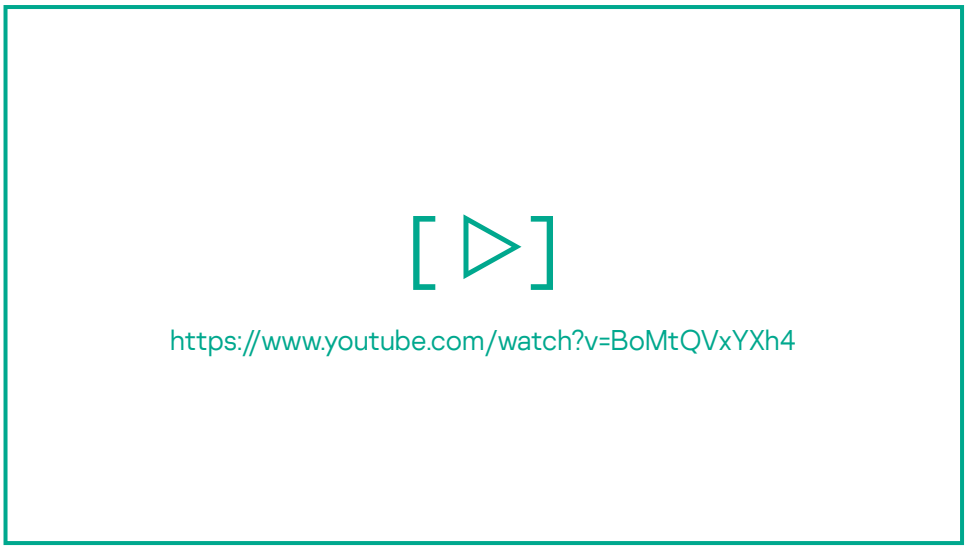
If you complete all the configuration steps but the users are not able to use their domain accounts for authentication in the KUMA web interface, it is recommended to check the configuration for the following issues:

- An email address is not indicated in the properties of the user account in Active Directory. If this is the case, an error message is displayed during the user's first authentication attempt and a KUMA account is not created.

- There is already an existing local KUMA account with the email address indicated in the domain account properties. If this is the case, the error message is displayed when the user attempts to perform authentication with the domain account.

- Domain authentication is disabled in the KUMA settings.

- An error occurred when entering the group of roles.
- The domain user name contains a space.



https://www.youtube.com/watch?v=BoMtQVxYXh4

Enabling and disabling domain authentication

Domain authentication is disabled by default. To enable it, enable the **State** toggle switch. If you want to temporarily suspend domain authentication after configuring a connection, you can disable it in the KUMA web interface without deleting the previously defined values of settings. If necessary, you can enable authentication again at any time.

To enable or disable domain authentication of users in the KUMA web interface:

1. In the solution web interface, select **Settings** → **Access** → **Domain authentication**.
2. In the **Authorization type** drop-down list, select one of the options:
   - **AD/ADFS**.
   - **FreeIPA**
3. Do one of the following:
   - If you want to enable domain authentication, enable the **State** toggle switch.
   - If you want to disable domain authentication, disable the **State** toggle switch.
4. Click the **Save** button.

   The selected settings will then be saved and applied.

Configuring connection between KUMA and FreeIPA

You can connect only to one FreeIPA domain. To do so, you must configure a connection to the domain controller.

To configure a connection to a FreeIPA domain controller:

1. In the solution web interface, select **Settings** → **Access** → **Domain authentication**.
2. In the **Authorization type** drop-down list, select **FreeIPA**.
3. Under **FreeIPA**, enable the **State** toggle switch.
4. In the **Base DN** field, enter the DistinguishedName of the root record to search for access groups in the FreeIPA catalog service. Record format: dc=example,dc=com.
5. In the **URL** field, indicate 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 up to three servers with domain controllers by separating them with commas. All of the specified servers must reside in the same domain.

6. If you want to use TLS encryption for the connection with the domain controller, select one of the following options from the **TLS mode** 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.

   - **LDAPS**.

     When using LDAPS, 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.

7. If TLS encryption is enabled, the **Secret** field becomes required and you must specify a secret of the 'certificate' type in that field. If you have previously uploaded a secret, select it from the **Secret drop-down list.** If necessary, click the + button to create a new secret of the 'certificate' type and select the secret from the drop-down list.
8. In the **Timeout in seconds** field, indicate the amount of time to wait for a response from the domain controller server. The default value is 0.

   If multiple addresses are specified in the **URL** field, KUMA waits for a response from the first server for the specified number of seconds. If no response is received during that time, the solution attempts to connect to the next server. If none of the indicated servers responds during the specified amount of time, the connection will be terminated with an error.

9. In the **Custom integration secret** drop-down list, select a secret with the 'credentials' type.

   If you want to upload a new secret of the 'credentials' type, click the + button on the right of the **Custom integration secret** drop-down list. This opens the **Secret** window; in that window, in the **Name** field, enter the name of the secret that will be displayed in the list after it is saved. In the **User** field, specify the DistinguishedName in the following format: uid=admin,cn=users,cn=accounts,dc=ipa,dc=test. Enter the **Password** and click **Save**.

The secret is uploaded and becomes available for selection in the **Custom integration secret** drop-down list.

10. If you want to configure domain authentication for a user with the KUMA general administrator role, use the **General administrators group** field to specify the DistinguishedName of the FreeIPA group containing the user. Additional roles for the General administrator are automatically activated in KUMA, therefore, you do not need to add them separately.

    In the case when multiple groups are specified for a user in the same tenant, the role with the highest-level permissions is used, with additional roles, if additional roles are assigned.

    Filter input example: `CN=KUMA team,OU=Groups,OU=Clients,DC=test,DC=domain`.

11. Click the **Save** button.

A connection with the FreeIPA domain controller is now configured.

You can also check the connection for the previously entered domain controller connection settings.

To check the connection to the domain controller:

1. In the solution web interface, select **Settings** → **Access** → **Domain authentication**.
2. In the **Authorization type** drop-down list, select **FreeIPA**.
3. In the **FreeIPA** settings block, select the relevant secret in the **User credentials** field.

   If necessary, you can create a new secret by clicking the + button or change the settings of an existing secret by clicking the button. If integration with FreeIPA is enabled, the secret selection is always reset when the page is loaded.
4. Click **Check**.

   After clicking the **Check** button, the system tests the connection with the domain and returns a notification with the test results. The system does not check if the users can log in or if the user group is configured correctly.

For domain authentication, add the groups for the KUMA user roles.

You can specify the groups only for the roles that require the configuration of domain authentication. You can leave the rest of the fields empty.

To add groups of user roles:

1. In the solution web interface, select **Settings** → **Access** → **Domain authentication**.
2. Under **Administration groups**, click **Add group**.
3. In the **Tenant** drop-down list, select the tenant of the users for whom you want to configure domain authentication.

4. In the **Roles** drop-down list, specify the roles for the user. You can select multiple roles. The following values are available:

   - **Tenant administrator**
   - **Tier 2 analyst**
   - **Tier 1 analyst**
   - **Junior analyst**
   - **Access to shared dashboards**. The role is only available when the Shared tenant is selected from the **Tenant** drop-down list. For the Shared tenant, this is the only role that can be added via the domain authentication settings.

   After you select the roles, a group filter field is displayed for each role. In the fields for each role, specify the full path to the domain group. The users of this domain group must have the capability to perform authentication with their domain accounts. Example of group specification: `CN=KUMA team`, `OU=Groups`, `OU=Clients`, `DC=test`, `DC=domain`.

   You can define a separate set of role filters for each tenant.

   If no filter is specified for a role, this means that conditions for creating an account through domain authentication are not specified for that role. Authentication with that role is impossible.

   After the first authentication under a domain account, domain user cards are created for users in the **Settings** → **Access** → **Users** section. For a domain user, the ability to change the main role (General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst) is blocked in the user card, while additional roles can be added or removed (Access to CII, Interaction with NCIRCC, Read shared resources), including management of additional role assignment to tenants. Roles assigned in the Domain authorization section and roles assigned in the user card supplement each other. For the General administrator, additional roles in KUMA are automatically activated, therefore you do not need to add them separately. If the General administrator role was assigned to a domain user, and the General administrator role was subsequently revoked, additional roles must be reassigned in the user card in the **Settings** → **Access** → **Users** section.

   > You can specify only one domain group for each role. If you want to specify multiple groups, you must repeat steps 2 to 4 for each group while specifying the same tenant.

5. If necessary, repeat steps 2–4 for each tenant for which you want to configure domain authentication with the following roles: Junior analyst, Tier 1 analyst, Tier 2 analyst, or Tenant administrator.

6. Click **Add**.

   The groups of user roles will be added. The defined settings will be applied the next time the user logs in to the KUMA web interface.

After the first authentication of the user, information about this user is displayed in the **Settings** → **Access** → **Users** section. The **Login** and **Password** fields received from the domain cannot be edited. The user role will also be unavailable for editing. To edit a role, you will have to change the user role groups. Changes to a role are applied after the next authentication of the user. The user continues working under the current role until the current session expires.

If the user name or email address is changed in the domain account properties, these changes must be manually made in the KUMA account.

Configuring connection between KUMA and Active Directory

You can connect only to one Active Directory domain. To do so, you must configure a connection to the domain controller.

To configure a connection to an Active Directory domain controller:

1. In the solution web interface, select **Settings** → **Access** → **Domain authentication**.
2. In the **Authorization type** drop-down list, select **AD/ADFS**.
3. Under **Active Directory (AD)**, enable the **State** toggle switch.
4. In the **Base DN** field, enter the DistinguishedName of the root record to search for access groups in the Active Directory catalog service.
5. In the **URL** field, indicate 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.

6. If you want to use TLS encryption for the connection with the domain controller, select one of the following options from the **TLS mode** 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.

   - **LDAPS**.

     When using LDAPS, 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.

7. If you enabled TLS encryption at the previous step, add a TLS certificate:

   - If you previously uploaded a certificate, select it from the **Secret** drop-down list.

     If no certificate was previously added, the drop-down list shows **No data**.

   - If you want to upload a new certificate, click the + button on the right of the **Secret** list. In the opened window, in the **Name** field, enter the name that will be displayed in the list of certificates after the certificate is added. Add the file containing the Active Directory certificate (X.509 certificate public keys in Base64 are supported) by clicking the **Upload certificate file** button. Click the **Save** button.

     The certificate will be uploaded and displayed in the **Secret** list.

8. 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 specified in the **URL** field, KUMA waits for a response from the first server for the specified number of seconds. If no response is received during that time, the solution attempts to connect to the next server. If none of the indicated servers responds during the specified amount of time, the connection will be terminated with an error.

9. To configure domain authentication for a user with the KUMA general administrator role, specify the DistinguishedName of the Active Directory group the user belongs to in the **General administrators group** field. Additional roles for the General administrator are automatically activated in KUMA, therefore you do not need to add them separately.

   > In the case when multiple groups are specified for a user in the same tenant, the role with the highest-level permissions is used, with additional roles, if additional roles are assigned.

   Filter input example: `CN=KUMA team,OU=Groups,OU=Clients,DC=test,DC=domain`.

10. Click the **Save** button.

A connection with the Active Directory domain controller is now configured.

You can also check the connection for the previously entered domain controller connection settings.

To check the connection to the domain controller:

1. In the solution web interface, select **Settings** → **Access** → **Domain authentication**.
2. In the **Authorization type** drop-down list, select **AD/ADFS**.
3. Under **Test connection**, select the relevant secret in the **User credentials** field.

   If necessary, you can create a new secret by clicking the + button or change the settings of an existing secret by clicking the button.

   The following formats for specifying a user are available in the **User** field: UserPrincipalName and domain\user.

4. Click **Check**.

   After clicking the **Check** button, the system tests the connection with the domain and returns a notification with the test results. The system does not check if the users can log in or if the user group is configured correctly.

For domain authentication, add the groups for the KUMA user roles.

> You can specify the groups only for the roles that require the configuration of domain authentication. You can leave the rest of the fields empty.

To add groups of user roles:

1. In the solution web interface, select **Settings** → **Access** → **Domain authentication**.
2. Under **Administration groups**, click **Add group**.
3. In the **Tenant** drop-down list, select the tenant of the users for whom you want to configure domain authentication.

4. In the **Roles** drop-down list, specify the roles for the user. You can select multiple roles. The following values are available:

   - **Tenant administrator**
   - **Tier 2 analyst**
   - **Tier 1 analyst**
   - **Junior analyst**
   - **Access to shared dashboards**. The role is only available when the Shared tenant is selected from the **Tenant** drop-down list. For the Shared tenant, this is the only role that can be added via the domain authentication settings.

   After you select the roles, a group filter field is displayed for each role. In the fields for each role, specify the full path to the domain group. The users of this domain group must have the capability to perform authentication with their domain accounts. Example of group specification: `CN=KUMA team`, `OU=Groups`, `OU=Clients`, `DC=test`, `DC=domain`.

   You can define a separate set of role filters for each tenant.

   If no filter is specified for a role, this means that conditions for creating an account through domain authentication are not specified for that role. Authentication with that role is impossible.

   After the first authentication under a domain account, domain user cards are created for users in the **Settings** → **Access** → **Users** section. For a domain user, the ability to change the main role (General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst) is blocked in the user card, while additional roles can be added or removed (Access to CII, Interaction with NCIRCC, Read shared resources), including management of additional role assignment to tenants. Roles assigned in the Domain authorization section and roles assigned in the user card supplement each other. For the General administrator, additional roles in KUMA are automatically activated, therefore you do not need to add them separately. If the General administrator role was assigned to a domain user, and the General administrator role was subsequently revoked, additional roles must be reassigned in the user card in the **Settings** → **Access** → **Users** section.

   > You can specify only one domain group for each role. If you want to specify multiple groups, you must repeat steps 2 to 4 for each group while specifying the same tenant.

5. If necessary, repeat steps 2–4 for each tenant for which you want to configure domain authentication with the following roles: Junior analyst, Tier 1 analyst, Tier 2 analyst, or Tenant administrator.

6. Click **Add**.

   The groups of user roles will be added. The defined settings will be applied the next time the user logs in to the KUMA web interface.

After the first authentication of the user, information about this user is displayed in the **Settings** → **Access** → **Users** section. The **Login** and **Password** fields received from the domain cannot be edited. The user role will also be unavailable for editing. To edit a role, you will have to change the user role groups. Changes to a role are applied after the next authentication of the user. The user continues working under the current role until the current session expires.

If the user name or email address is changed in the domain account properties, these changes must be manually made in the KUMA account.

Configuring connection between KUMA and Active Directory Federation Services

To configure domain authentication in KUMA and ensure that users can log in to KUMA using their accounts without specifying a user name and password, first create a connection group and configure the rules in AD FS or make sure that the necessary connection groups and rules already exist.

After configuration, the **Sign in via ADFS** button appears on the KUMA login page.

The **Sign in via ADFS** button is hidden on the KUMA login page in the following conditions:

- The **FreeIPA** option is selected in the **Authorization type** drop-down list.
- The **AD/ADFS** option is selected in the **Authorization type** drop-down list and the settings for AD FS are not specified or the **State** toggle switch is enabled for AD FS settings.

You can connect only to one ADFS domain. To do so, you must configure a connection to the domain controller.

To configure a connection to an AD FS domain controller:

1. In the solution web interface, select **Settings** → **Access** → **Domain authentication**.
2. In the **Authorization type** drop-down list, select **AD/ADFS**.
3. Under **Active Directory Federation Services (ADFS)**, enable the **State** toggle switch.
4. In the **Client ID** field, enter the KUMA ID from the **Client ID** field in AD FS.
5. In the **Relying party identifier** field, enter the KUMA ID from the **Relying party identifiers** field in the AD FS.
6. Enter the **Connect Metadata URI** from the **Connect Metadata URI** field. The URI consists of the host of the AD FS (https://adfs.example.com) and the endpoint configuration parameter (/adfs/.well-known/openid-configuration).

   For example, https://adfs.example.com/adfs/.well-known/openid-configuration.
7. Enter the **Redirect URL** from the **Redirect URL** field in the AD FS. The value of the **Redirect URL** field in the AD FS is defined when the Application group is configured. In the AD FS, you need to indicate the KUMA FQDN and the </sso-callback> substring. In KUMA, the URL must be indicated without the substring, for example: https://kuma-example:7220/

   In a distributed installation with KUMA Core in a Raft cluster, you must configure the balancer and then specify the FQDN of the balancer. For example, in the AD FS settings, you need to specify https://kuma-load-balancer.com:7220</sso-callback>, and in KUMA, in the **Redirect URL** field, you need to specify https://kuma-load-balancer.com:7220
8. If you want to configure domain authentication for a user with the KUMA general administrator role, use the **General administrators group** field to specify the DistinguishedName of the Active Directory Federation Services group containing the user. Additional roles for the General administrator are automatically activated in KUMA, therefore, you do not need to add them separately.

   In the case when multiple groups are specified for a user in the same tenant, the role with the highest-level rights is used, with additional rights, if additional roles are assigned.

Filter input example: `CN=KUMA team,OU=Groups,OU=Clients,DC=test,DC=domain`.

9. Click the **Save** button.

A connection with the Active Directory Federation Services domain controller is now configured.

If, when trying to log in to KUMA via ADFS, the user gets an `Access denied` pop-up message, click the **Reset certificate** button. A new certificate will be generated automatically.

For domain authentication, add the groups for the KUMA user roles.

You can specify the groups only for the roles that require the configuration of domain authentication. You can leave the rest of the fields empty.

To add groups of user roles:

1. In the solution web interface, select **Settings** → **Access** → **Domain authentication**.
2. Under **Administration groups**, click **Add group**.
3. In the **Tenant** drop-down list, select the tenant of the users for whom you want to configure domain authentication.
4. In the **Roles** drop-down list, specify the roles for the user. You can select multiple roles. The following values are available:
   - **Tenant administrator**
   - **Tier 2 analyst**
   - **Tier 1 analyst**
   - **Junior analyst**
   - **Access to shared dashboards**. The role is only available when the Shared tenant is selected from the **Tenant** drop-down list. For the Shared tenant, this is the only role that can be added via the domain authentication settings.

   After you select the roles, a group filter field is displayed for each role. In the fields for each role, specify the full path to the domain group. The users of this domain group must have the capability to perform authentication with their domain accounts. Example of group specification: `CN=KUMA team`, `OU=Groups`, `OU=Clients`, `DC=test`, `DC=domain`.

   You can define a separate set of role filters for each tenant.

   If no filter is specified for a role, this means that conditions for creating an account through domain authentication are not specified for that role. Authentication with that role is impossible.

After the first authentication under a domain account, domain user cards are created for users in the **Settings** → **Access** → **Users** section. For a domain user, the ability to change the main role (General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst) is blocked in the user card, while additional roles can be added or removed (Access to CII, Interaction with NCIRCC, Read shared resources), including management of additional role assignment to tenants. Roles assigned in the Domain authorization section and roles assigned in the user card supplement each other. For the General administrator, additional roles in KUMA are automatically activated, therefore you do not need to add them separately. If the General administrator role was assigned to a domain user, and the General administrator role was subsequently revoked, additional roles must be reassigned in the user card in the **Settings** → **Access** → **Users** section.

> You can specify only one domain group for each role. If you want to specify multiple groups, you must repeat steps 2 to 4 for each group while specifying the same tenant.

5. If necessary, repeat steps 2–4 for each tenant for which you want to configure domain authentication with the following roles: Junior analyst, Tier 1 analyst, Tier 2 analyst, or Tenant administrator.

6. Click **Add**.

The groups of user roles will be added. The defined settings will be applied the next time the user logs in to the KUMA web interface.

After the first authentication of the user, information about this user is displayed in the **Settings** → **Access** → **Users** section. The **Login** and **Password** fields received from the domain cannot be edited. The user role will also be unavailable for editing. To edit a role, you will have to change the user role groups. Changes to a role are applied after the next authentication of the user. The user continues working under the current role until the current session expires.

If the user name or email address is changed in the domain account properties, these changes must be manually made in the KUMA account.

Configuring connection in Active Directory Federation Services

This section provides instructions on how to create a new connection group and configure rules for the created connection group in Active Directory Federation Services (AD FS).

The AD FS role must already be configured on the server.

Creating a new connection group

1. In **Server Manager**, in the **Tools** menu, select **ADFS Management**.

   In ADFS, select the **Application groups** section and in the **Actions** section click **Add Application Group**.

2. In the **Add Application Group Wizard** window that opens, in the **Welcome** section **Name** field, specify the name of the new connection group. Example: new-application-group.

   In the **Template** field, in the **Client-Server applications** group, select **Native application accessing a web API**.

   Click **Next** to proceed to the next step of creating and configuring a connection group.

3. In the **Native application** section that opens, **Name** and **Client Identifier** fields are filled in automatically.

   Specify the value of the **Client Identifier** field in the **Client Identifier** field in KUMA, when configuring domain authentication.

   In the **Redirect URI** field, enter the URI for redirection from AD FS with the `/sso-callback` substring, and click **Add**. Example: `https://adfs.example.com:7220/sso-callback`.

   Click **Next** to proceed to the next configuration step.

4. In the **Configure Web API** section that opens, in the **Identifiers** field, add the trusted party ID and click **Add**. It can be any arbitrary value. Example: `test-demo`.

   Specify the value of the **Identifier** field in KUMA, in the **Relying party identifiers** field, when configuring domain authentication.

   Click **Next** to proceed to the next configuration step.

5. In the **Apply Access Control Policy** section that opens, select the **Permit everyone** policy value.

   Click **Next** to proceed to the next configuration step.

6. In the **Configure Application Permissions** section that opens, the **Client application** field is filled in automatically.

   In the **Permitted scopes** field, select the check box for **allatclaims** and **openid** options.

   Click **Next** to proceed to the next configuration step.

7. In the **Summary** section that opens, check the settings.

   If the settings are correct and you are ready to add a group, click **Next**.

This will add a new group. You can proceed to configure the rules for the created group.

Adding rules for a connection group

1. In **Server Manager**, in the **Tools** menu, select **ADFS Management**.

   In AD FS, select the **Application groups** section, and select the required connection group from the list in the window that opens. Example: `new-application-group`.

2. In the **Application groups** window, in the **Actions** section, click **Properties**.

   In the **new-application-group Properties** window that opens, in the **Applications** section, double-click **new-application-group - Web API**.

   In the **new-application-group - Web API Properties** window that opens, switch to the **Issuance Transform Rules** tab and click **Add rule**.

   In the **Add Transform Claim Rule Wizard** window that opens, in the **Choose Rule Type** section, select **Send LDAP Attributes as Claims** from the drop-down list.

   Click **Next** to proceed to the next configuration step.

3. In the **Configure Claim Rule** section, specify the rule name in the **Claim rule name** field. Example: `rule-name-01`.

   In the **Attribute store** drop-down list, select **Active directory**.

   In the **Mapping of LDAP attributes to outgoing claim types** field, map the following fields:

LDAP Attribute	Outgoing Claim Type
`User-Principal-Name`	`UserPrincipalName`
`Display-Name`	`displayName`
`E-Mail-Addresses`	`Mail`
`Is-Member-Of-DL`	`MemberOf`

   Click **Finish** to complete the configuration.

4. Return to the **new-application-group - Web API Properties** window, switch to the **Issuance Transform Rules** tab and click **Add rule**. In the **Add Transform Claim Rule Wizard** window that opens, in the **Choose Rule Type** section, select **Send claims using a custom rule** from the drop-down list.

   Click **Finish** to continue the configuration.

5. In the **Configure Claim Rule** section, specify the rule name in the **Claims rule name** field. Example: `rule-name-02`.

   In the **Custom rule** field, specify the following settings:

   ```
   c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD
   AUTHORITY"]=> issue(store = "Active Directory", types = ("ObjectGUID"), query = ";ObjectGUID;{0}", param =
   c.Value);
   ```

   Click **Finish** to complete the configuration.

6. The system proceeds to the **new-application-group - Web API Properties** window and the **Issuance Transform Rules** tab.

   To apply the rules, on the **Issuance Transform Rules** tab that opens, click **Apply** or **OK**.

The configuration of groups and rules in AD FS is complete. You can proceed to configure domain authentication in KUMA.

Fixing the 'Access denied' error when logging in to KUMA via ADFS

When you try to log in to KUMA using ADFS, the `Access denied` or `Insufficient rights` pop-up message may appear. The KUMA Core log shows the `Data source certificate has been changed` error.

This error indicates that the ADFS certificate has been changed. To fix the error and resume domain authentication, in domain controller connection settings, click the **Reset certificate** button. A new certificate will be generated automatically.

NCIRCC integration

In the KUMA web interface, you can create a connection to the National Computer Incident Response & Coordination Center Incidents (hereinafter referred to as "NCIRCC"). This will let you export incidents registered by KUMA to NCIRCC. Integration is configured under **Settings** → **NCIRCC** in the KUMA web interface. All fields that you fill out in the settings section are automatically sent to the NCIRCC data submission form.

Data in KUMA and NCIRCC is synchronized every 5-10 minutes.

To create a connection to NCIRCC:

1. In the KUMA web interface, open the **Settings** → **Integrations** → **NCIRCC** section.
2. Use the **Status** toggle switch to enable the integration. The integration is disabled by default.
3. In the **URL** field, enter the URL for accessing NCIRCC.
4. In the **Tenant** drop-down list, select a tenant for messages from NCIRCC.

5. In the **Token** drop-down list, create or select an existing secret with the API token that was issued to your organization for connecting to NCIRCC:

   - If you already have a secret, you can select it from the drop-down list.
   - If you want to create a new secret:

     a. Click **Create** and in the **Create secret** window, specify the following settings:

        - **Name** (required)—unique name of the resource you are creating. The name must contain 1 to 128 Unicode characters.
        - **Token** (required)—token that was issued to your organization for a connection to NCIRCC.
        - **Description**—service description: up to 256 Unicode characters.

     b. If necessary, select the tags for the integration from the **Tags** drop-down list.

     c. Click **Create**.

     The secret containing the token for connecting to NCIRCC will be created. It is stored in the **Resources** → **Resources** → **Secrets** section and is owned by the main tenant.

6. In the **Company scope** drop-down list, select the required value.
7. In the **Company name** field, specify the name of the company for which you are configuring the integration.
8. In the **Location** drop-down list, specify the location of your company.
9. In the **Root CA** drop-down list, create a secret or select an existing secret:

   - If you already have a secret, you can select it from the drop-down list.
   - If you want to create a new secret:

     a. Click the **Create** button and specify the following settings:

        - **Name** is the unique name for the resource you are creating. The name must contain 1 to 128 Unicode characters.
        - **Type** is the type of the secret (certificate).
        - For the **Certificate file**, click the **Upload certificate** button and select the certificate of the intermediate certification authority that is uploaded and installed on the KUMA Core server.

          Download and install the certificate of the intermediate certification authority.

To install and trust the certificate of the intermediate certification authority on the KUMA Core server:

1. Follow the NCIRCC account link. For example, https://lk.cert.gov.ru.
2. Right-click to call up the **View site details** context menu to the left of the link in the address bar.
   - If you are using an encrypted connection, in the context menu, select **Connection is secure** and in the drop-down list under **Certificate is valid**, click the **Show certificate** link.
   - If you are using an unencrypted connection, in the menu, click **Certificate details**.

   Depending on your browser, the position of the menu and the order of items may differ.
3. In the displayed **Certificate Viewer** window, under **Issued By**, in the **Common Name (CN)** field you can find the name of the certificate that you need. For example, GlobalSign GCC R6 AlphaSSL CA 2023.

   Remember the name of the certificate because you will need to download it at the next step.
4. Click the https://support.globalsign.com/ca-certificates/intermediate-certificates/alphassl-intermediate-certificates link, find the certificate from step 3, and click "View as BASE64".
5. Paste the displayed certificate strings into a file and add the file with the certificate strings as the secret in KUMA.
6. After installing the certificate, restart the KUMA Core server.

As a result, the certificate is installed and you can proceed with configuring the integration.

- **Tags**—tags for the certificate.
- **Description**—service description: up to 256 Unicode characters.

b. Click **Create**.

The secret with the certificate of the intermediate certification authority is created. It is stored in the **Resources** → **Resources** → **Secrets** section and is owned by the main tenant.

10. If necessary, under **Proxy**, create or select an existing proxy server that must be used when connecting to NCIRCC.
11. Specify the NCIRCC response timeout in seconds.
12. If necessary, specify the tax ID (INN), operator address, and email address for sending notification information.
13. Click **Save**.

KUMA is now integrated with NCIRCC. Now you can export incidents to it. You can click the **Test connection** button to make sure that a connection with NCIRCC is established.

You can use the **Disabled** check box to enable or disable integration.

Possible errors

If the *https://lk.cert.gov.ru/api/v2/incidents? x509: certificate signed by unknown authority* error is returned when you configure integration with NCIRCC, **download and install the certificate of the intermediate certification authority on the KUMA Core server**.

To install and trust the certificate of the intermediate certification authority on the KUMA Core server:

1. Follow the NCIRCC account link. For example, https://lk.cert.gov.ru.
2. Right-click to call up the **View site details** context menu to the left of the link in the address bar.
   - If you are using an encrypted connection, in the context menu, select **Connection is secure** and in the drop-down list under **Certificate is valid**, click the **Show certificate** link.
   - If you are using an unencrypted connection, in the menu, click **Certificate details**.

   Depending on your browser, the position of the menu and the order of items may differ.
3. In the displayed **Certificate Viewer** window, under **Issued By**, in the **Common Name (CN)** field you can find the name of the certificate that you need. For example, GlobalSign GCC R6 AlphaSSL CA 2023.

   Remember the name of the certificate because you will need to download it at the next step.
4. Click the https://support.globalsign.com/ca-certificates/intermediate-certificates/alphassl-intermediate-certificates link, find the certificate from step 3, and click "View as BASE64".
5. Paste the displayed certificate strings into a file and add the file with the certificate strings as the secret in KUMA.
6. After installing the certificate, restart the KUMA Core server.

As a result, the certificate is installed and you can proceed with configuring the integration.

Integration with the Security Orchestration Automation and Response Platform (SOAR)

Security Orchestration, Automation and Response Platform (hereinafter referred to as SOAR) is a software platform used for automation of monitoring, processing, and responding to information security incidents. It aggregates cyberthreat data from various sources into a single database for further analysis and investigation to facilitate incident response capabilities.

SOAR can be integrated with KUMA. After configuring integration, you can perform the following tasks in SOAR:

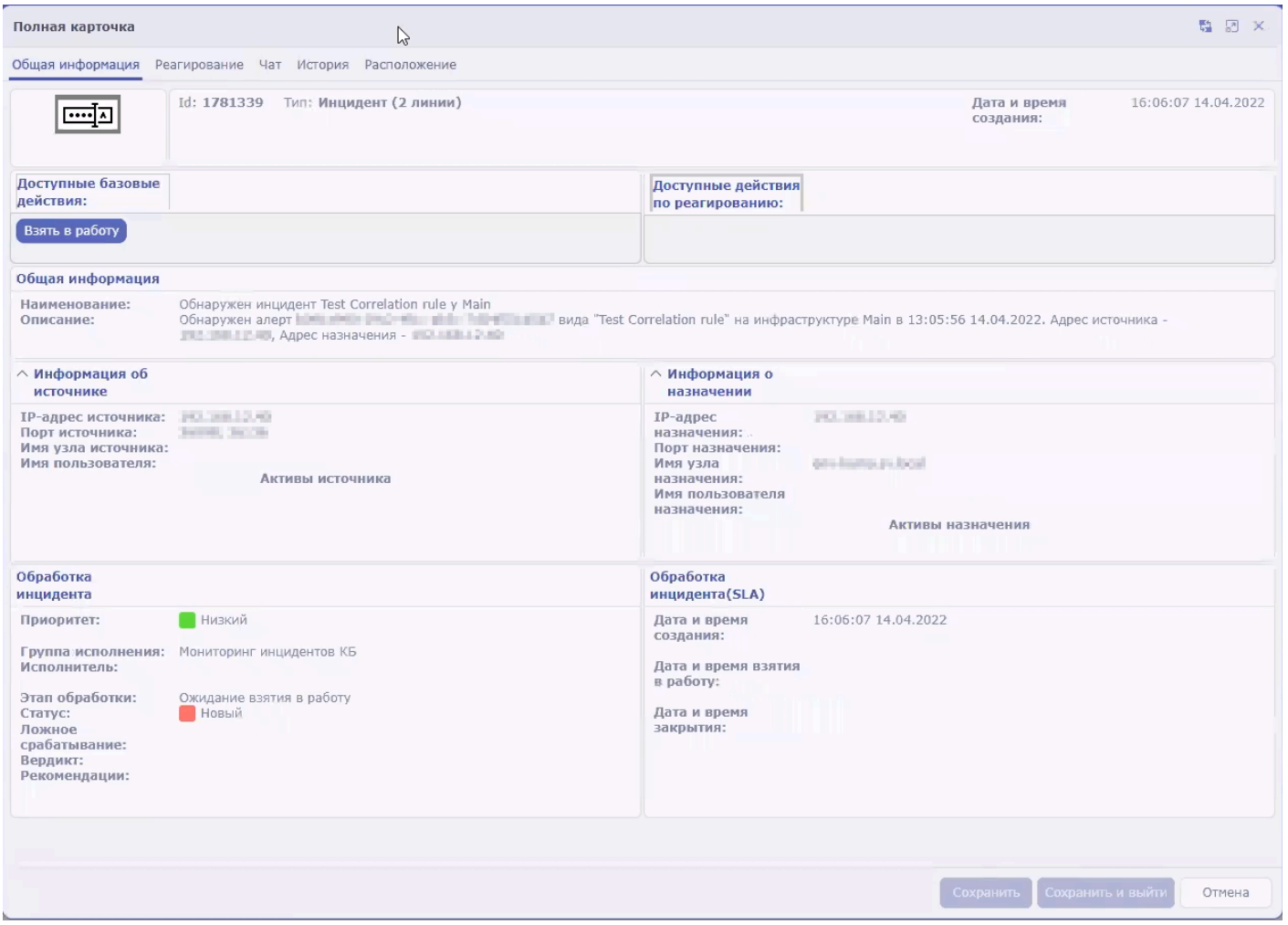
- Request information about alerts from KUMA. In SOAR, *incidents* are created based on received data.
- Send requests to KUMA to close alerts.

Integration is implemented by using the KUMA REST API. On the Security Vision IRP side, integration is carried out by using the preconfigured **Kaspersky KUMA** connector. Contact your SOAR vendor to learn more about the methods and conditions for obtaining a **Kaspersky KUMA** connector.

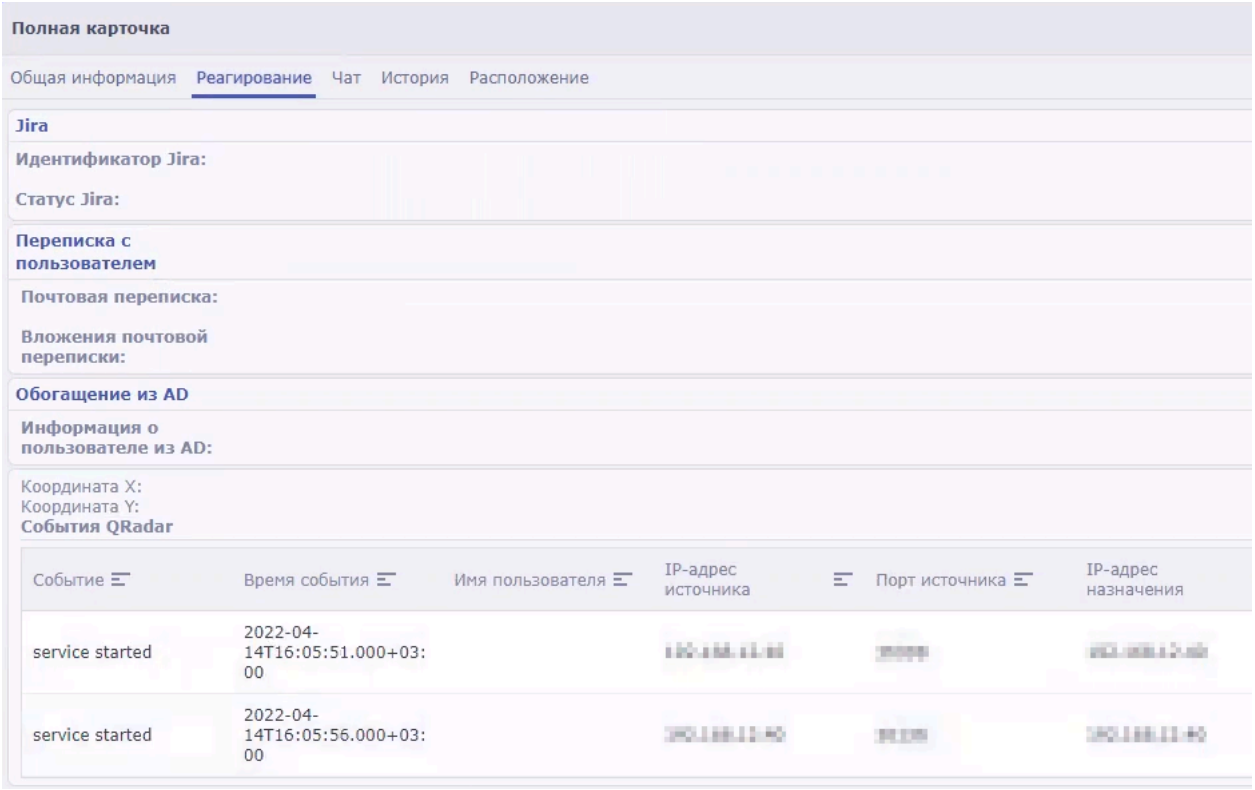
Managing SOAR incidents

SOAR incidents generated from KUMA alert data can be viewed in SOAR under **Incidents** → **Incidents (2 lines)** → **All incidents (2 lines)**. Events related to KUMA alerts are logged in each SOAR incident. Imported events can be viewed on the **Response** tab.

KUMA alert imported into SOAR as an incident



Security Vision IRP incident that was created based on KUMA alert



Events from KUMA alert that were imported to Security Vision IRP

Configuring integration in KUMA

To configure KUMA integration with SOAR, you must configure authorization of API requests in KUMA. To do so, you need to create a token for the KUMA user on whose behalf the API requests will be processed on KUMA side.

A token can be generated in your account profile. Users with the General Administrator role can generate tokens in the accounts of other users. You can always generate a new token.

To generate a token in your account profile:

1. In the KUMA web interface, click the user account name in the lower-left corner of the window and click the **Profile** button in the opened menu.

   The **User** window with your user account parameters opens.

2. Click the **Generate token** button.

3. Copy the generated token displayed in the opened window. You will need it to configure SOAR.

   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.

The generated token must be specified in the SOAR connector settings.

Configuring integration in SOAR

Configuration of integration in SOAR consists of importing and configuring a connector. If necessary, you can also change other SOAR settings related to KUMA data processing, such as the data processing schedule and worker.

For more detailed information about configuring SOAR, please refer to the product documentation.

Importing and configuring a connector

Adding a connector to SOAR

Integration of SOAR and KUMA is performed using the **Kaspersky KUMA** connector. Contact your SOAR vendor to learn more about the methods and conditions for obtaining a **Kaspersky KUMA** connector.

*To import the **Kaspersky KUMA** connector to SOAR:*

1. In SOAR, open the **Settings** → **Connectors** → **Connectors** section.

   A list of connectors added to SOAR is displayed.

2. At the top of the screen, click the import button and select the ZIP archive containing the **Kaspersky KUMA** connector.

The connector is imported into SOAR and is ready to be configured.

Configuring a connector for a connection to KUMA

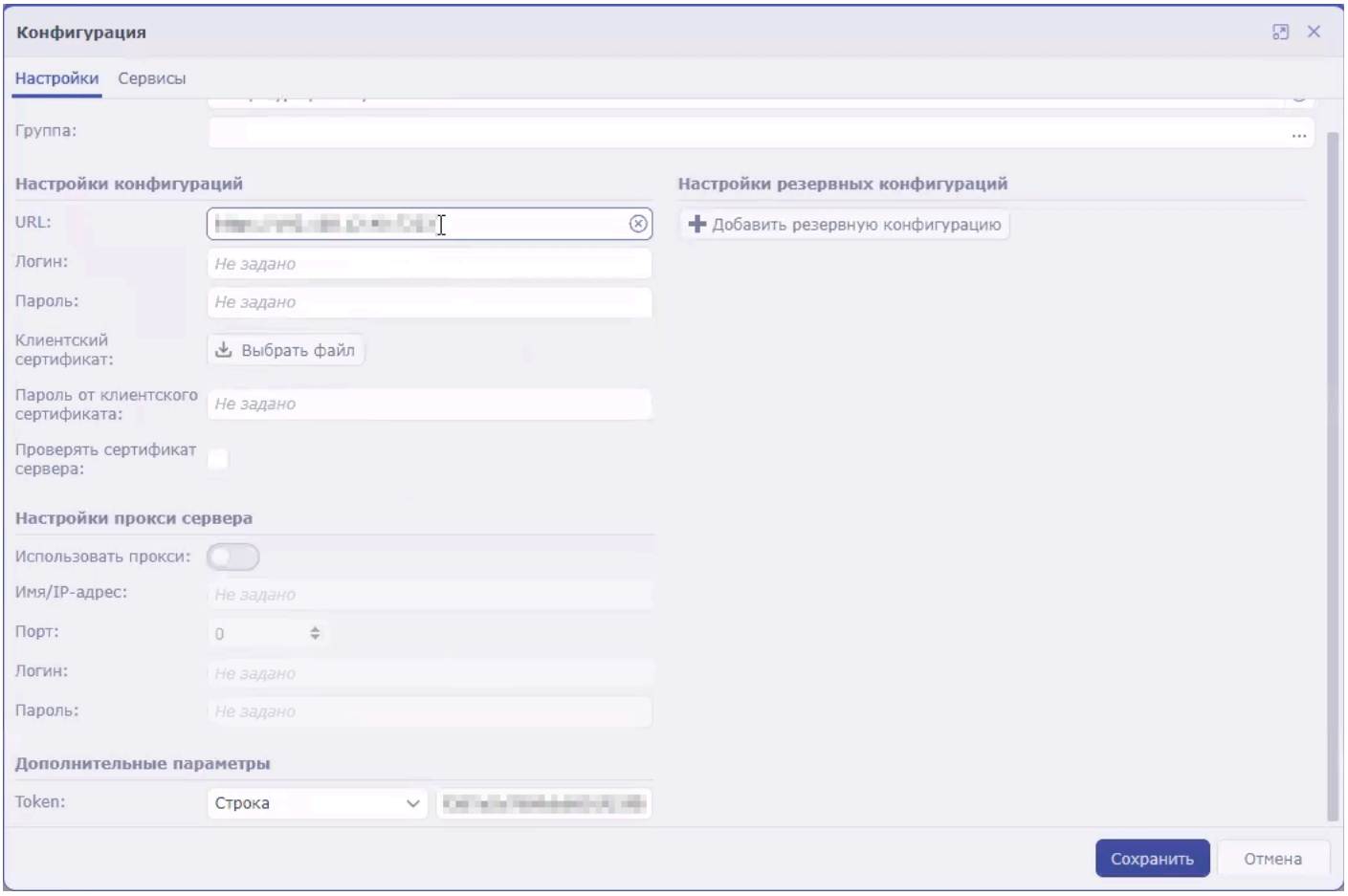
To use a connector, you need to configure its connection to KUMA.

*To configure a connection to KUMA in SOAR using the **Kaspersky KUMA** connector:*

1. In SOAR, open the **Settings** → **Connectors** → **Connectors** section.
   A list of connectors added to your SOAR is displayed.
2. Select the **Kaspersky KUMA** connector.
   The general settings of the connector will be displayed.
3. Under **Connector settings**, click the **Edit** button.
   The connector configuration will be displayed.
4. In the **URL** field, specify the address and port of KUMA. For example, `kuma.example.com:7223`.
5. In the **Token** field, specify KUMA user API token.

The connection to KUMA is configured in the SOAR connector (click the links below for an image).

Security Vision IRP connector settings



Security Vision IRP connector settings

Configuring commands for interaction with KUMA in the SOAR connector

You can use SOAR to receive information about KUMA alerts (referred to as *incidents* in SOAR terminology) and send requests to close these alerts. To perform these actions, you need to configure the appropriate commands in the SOAR connector.

The instructions below describe how to add commands to receive and close alerts. However, if you need to implement more complex logic of interaction between SOAR and KUMA, you can similarly create your own commands containing other API requests.

To configure a command to receive alert information from KUMA:

1. In SOAR, open the **Settings** → **Connectors** → **Connectors** section.

   A list of connectors added to SOAR is displayed.

2. Select the **Kaspersky KUMA** connector.

   The general settings of the connector will be displayed.

3. Click the **+Command** button.

   The command creation window opens.

4. Specify the command settings for receiving alerts:

   - In the **Name** field, enter the command name: `Receive incidents`.
   - In the **Request type** drop-down list, select **GET**.
   - In the **Called method** field, enter the API request to search for alerts:

     `api/v1/alerts/?withEvents&status=new`
   - Under **Request headers**, in the **Name** field, indicate `authorization`. In the **Value** field, indicate **Bearer <token>**.
   - In the **Content type** drop-down list, select **application/json**.

5. Save the command and close the window.

The connector command is configured. When this command is executed, the SOAR connector queries KUMA for information about all alerts with the **New** status and all events related to those alerts. The received data is sent to the SOAR processor, which uses it to create SOAR incidents. If new data appears in an alert that has been already imported into SOAR, incident information is updated in SOAR.

To configure a command to close KUMA alerts:

1. In SOAR, open the **Settings** → **Connectors** → **Connectors** section.

   A list of connectors added to SOAR is displayed.

2. Select the **Kaspersky KUMA** connector.

   The general settings of the connector will be displayed.

3. Click the **+Command** button.

   The command creation window will be displayed.

4. Specify the command settings for receiving alerts:

   - In the **Name** field, enter the command name: `Close incident`.
   - In the **Request type** drop-down list, select **POST**.
   - In the **Called method** field, enter API request to close an alert:

     `api/v1/alerts/close`
   - In the **Request** field, enter the contents of the sent API request:

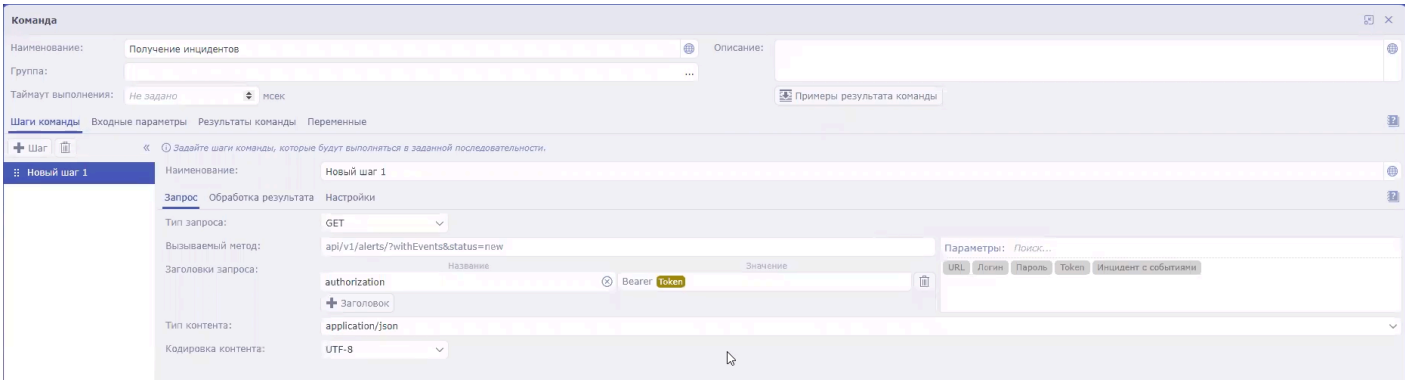
     `{"id":"<Alert ID>","reason":"responded"}`

     You can create multiple commands for different reasons to close alerts, such as responded, incorrect data, and incorrect correlation rule.
   - Under **Request headers**, in the **Name** field, indicate `authorization`. In the **Value** field, indicate **Bearer <token>**.
   - In the **Content type** drop-down list, select **application/json**.

5. Save the command and close the window.

The connector command is configured (click the links below for an image). When this command is executed, the incident is closed in SOAR and the corresponding alert is closed in KUMA.

Creating commands in SOAR



Creating commands in SOAR

After the SOAR connector is configured, KUMA alerts are sent to the platform as SOAR incidents. Then you need to configure incident handling in SOAR based on the security policies of your organization.

Configuring the handler, schedule, and worker process

SOAR handler

The SOAR handler receives information about KUMA alerts from the SOAR connector and uses the information to create SOAR incidents. A predefined **KUMA (Incidents)** handler is used for processing data. The settings of the **KUMA (Incidents)** handler are available in SOAR under **Settings** → **Event processing** → **Event handlers**:

- You can view the rules for processing KUMA alerts in the handler settings on the **Normalization** tab.
- You can view the actions available when creating new objects in the handler settings on the **Actions** tab for creating objects of the **Incident (2 lines)** type.

Handler run schedule

The connector and handler are started according to a predefined **KUMA** schedule. This schedule can be configured in SOAR under **Settings** → **Event processing** → **Schedule**:

- Under **Connector settings**, you can configure the settings for starting the connector.
- Under **Handler settings**, you can configure the settings for starting the handler.

SOAR workflow

The life cycle of SOAR incidents created based on KUMA alerts follows the preconfigured **Incident processing (2 lines)** worker. The worker can be configured in SOAR under **Settings** → **Workers** → **Worker templates**: select the **Incident processing (2 lines)** worker and click the transaction or state that you need to change.

Integration with KICS for Networks and KATA

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.

Kaspersky Anti Targeted Attack (hereinafter referred to as KATA) is an application designed for protection of corporate IT infrastructure and early detection of threats such as zero-day attacks, targeted attacks, and advanced persistent threats.

KICS for Networks or KATA can be integrated with KUMA. After configuring integration, you can perform the following tasks in KUMA:

- Import asset information from KICS for Networks or KATA to KUMA.

  Import of asset information from KATA version 7.1 is available only in KUMA version 4.0 and later.
- Send asset status change commands from KUMA to KICS for Networks or KATA.

Unlike KUMA, KICS for Networks and KATA refers to assets as devices.

The integration of KICS for Networks or KATA and KUMA must be configured on both sides:

1. In KICS for Networks or KATA, you need to create a KUMA connector and save the communication data package of this connector.

   You can configure the integration on the side of KICS for Networks 4.0 or later as well as on the side of KATA 7.0 and later versions. Configuring the integration on either side allows correctly transmitting asset and event information to KUMA.

2. In KUMA, the communication data package of the connector is used to create a connection to KICS for Networks or KATA.

The integration described in this section applies to importing asset information. KICS for Networks or KATA can also be configured to send events to KUMA. To do so, you need to create a SIEM/Syslog connector in KICS for Networks or KATA, and configure a collector on the KUMA side.

Configuring integration in KICS for Networks or KATA

Integration can be configured on the side of KICS for Networks version 4.0 or on the side of KATA 7.0 or a later version For the import of data and events to work correctly, you must use the KUMA connector and the communication data package created in the appropriate system.

We recommend configuring integration of KICS for Networks or KATA with KUMA after ending Process Control rules learning mode. For more details, please refer to the KICS for Networks documentation.

On the KICS for Networks or KATA side, integration configuration consists of creating a *KUMA-type connector*. In KICS for Networks and KATA, connectors are specialized application modules that enable KICS for Networks or KATA to exchange data with recipient systems, including KUMA. For more details on creating connectors, please refer to the KICS for Networks documentation.

When a connector is added to KICS for Networks or KATA, a *communication data package* is automatically created for this connector. A communication data package is an encrypted configuration file for connecting to KICS for Networks or KATA that is used when configuring integration on the KUMA side.

Configuring integration in KUMA

We recommend configuring integration of KICS for Networks or KATA with KUMA after ending Process Control rules learning mode. For more details, please refer to the KICS for Networks documentation.

To configure integration with KICS for Networks or KATA in KUMA:

1. In the KUMA web interface, open the **Settings** → **Integrations** → **KICS/KATA** section.

   This opens the **KICS/KATA integration by tenant** window.

2. Click the **Add settings for a new tenant** button.

3. This opens a window; in that window, specify the following KICS for Networks or KATA integration settings:

   - Turn on the **State** toggle switch. The toggle switch is turned on by default.
   - In the **Tenant** drop-down list, select a tenant for the integration with KICS for Networks or KATA.
   - Enable the **Enable responses** toggle switch to be able to change the statuses of KICS for Networks or KATA assets using KUMA response rules.

4. In the area with the **Connections** table, add a new connection by clicking the **+ Add** button.

5. This opens the **Create connection** window; in that window, manage the following settings:

   - Turn on the **State** toggle switch. The toggle switch is turned on by default.
   - In the **Communication data package** field, click the add (+) icon and select the communication data package of the connector that was created in KICS for Networks or KATA when creating a connector for the KUMA connection.
   - In the **Communication data package password** field, enter the password set for the communication data package.
   - In the **API version** field, specify the API version used by KICS for Networks or KATA. The default value is `Auto` (version 4.0).

6. Click **Save**.

   The created connection is automatically displayed in the table of connections of the selected tenant.

7. If necessary, add more connections within the same tenant.

8. In the **KICS/KATA integration by tenant** window, click **Add**.

Integration with KICS for Networks or KATA is configured in KUMA. You can view the ID of the node on which the KICS for Networks or KATA connector will run by clicking the link with its IP address.

Enabling and disabling integration with KICS for Networks or KATA

To enable or disable KICS for Networks or KATA integration for a tenant:

1. In the KUMA web interface, open the **Settings** → **Integrations** → **KICS/KATA** section.

   This opens the **KICS/KATA integration by tenant** window.

2. Do one of the following:

   - Use the **Status** toggle switch to enable or disable the integration for the tenant.
   - Click the **Add settings for a new tenant** button to open a window and in that window, use the **State** toggle switch to enable or disable the integration for the tenant, and then save the changes.

Changing the data update frequency

KUMA queries KICS for Networks or KATA 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 or KATA, a task is created in the **Task manager** section of the KUMA web interface.

To edit the schedule for importing information about KICS for Networks or KATA assets:

1. In the KUMA web interface, open the **Settings** → **Integrations** → **KICS/KATA** section.
2. Select the relevant tenant.

   This opens the **KICS/KATA integration by tenant** window.
3. In the **Data refresh interval in hours** field, specify the data update interval in hours. The default value is 3.
4. Save your changes.

The import schedule has been changed.

Special considerations when importing asset information from KICS for Networks or KATA

Regardless of which side the integration is initiated from (KICS for Networks or KATA), the import of assets and vulnerabilities into KUMA is performed in accordance with standard rules, and IDs are preserved.

Importing assets

Assets are imported according to the asset import rules. Only assets with the **Authorized** and **Unauthorized** statuses are imported.

KICS for Networks and KATA assets are identified by a combination of the following parameters:

- IP address of the KICS for Networks or KATA instance with which the integration is configured.
- KICS for Networks or KATA connector ID is used to configure the integration.
- ID assigned to the asset (or "device") in the KICS for Networks or KATA instance.

Importing vulnerability information

When importing assets, KUMA also receives information about active vulnerabilities in KICS for Networks or KATA. If a vulnerability has been flagged as Remediated or Negligible in KICS for Networks or KATA, 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 or KATA in the **Asset details** window in the **Vulnerabilities** settings block.

> In KICS for Networks or KATA, vulnerabilities are referred to as *risks* and are classified 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 or KATA, the asset is deleted after 30 days.

Changing the status of a KICS for Networks or KATA asset

After configuring integration, you can change the statuses of KICS for Networks or KATA 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 or KATA.

Manually changing the status of a KICS for Networks or KATA asset

Users with the General administrator, Tenant administrator, and Tier 2 analyst roles can manually change the statuses of assets imported from KICS for Networks or KATA in the tenants available to them.

To manually change a KICS for Networks or KATA asset status:

1. In the **Assets** section of the KUMA web interface, click the asset that you want to edit.

   The **Asset details** area opens in the right part of the window.

2. In the **Status in KICS/KATA** drop-down list, select the status that you want to assign to the KICS for Networks or KATA asset. The *Authorized* or *Unauthorized* statuses are available.

The asset status is changed. The new status is displayed in KICS for Networks or KATA and in KUMA.

Automatically changing the status of a KICS for Networks or KATA asset

Automatic changes to the statuses of KICS for Networks or KATA assets are implemented using response rules. The rules must be added to the correlator, which will determine the conditions for triggering these rules.

Kaspersky Automated Security Awareness Platform

Kaspersky Automated Security Awareness Platform (hereinafter also referred to as "ASAP") 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.

ASAP 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 ASAP and KUMA includes configuring API connection to ASAP. The process takes place in both solutions:

1. In ASAP, create an authorization token and obtain an address for API requests.
2. In KUMA, specify the address for API requests in ASAP, add an authorization token for API requests, and specify the email address of the ASAP administrator to receive notifications.

Creating a token in ASAP and getting a link for API requests

In order to be authorized, the API requests from KUMA to ASAP must be signed by a token created in ASAP. Only the company administrators can create tokens.

Creating a token

To create a token:

1. Sign in to the ASAP web interface.
2. In the **Control panel** section, click **Import and synchronization**, 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 ASAP 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 ASAP for API requests:

1. Sign in to the ASAP web interface.

2. In the **Control panel** section, click **Import and synchronization**, and then open the **Open API** tab.

3. A link for accessing ASAP 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 ASAP:

1. In the KUMA web interface, open **Settings** → **Integrations** → **Kaspersky Automated Security Awareness Platform** section.

   The **Kaspersky Automated Security Awareness Platform** window opens.

2. Make sure the **Status** toggle switch is enabled and the integration is enabled.

3. In the **Secret** drop-down list, select a secret to use for the Kaspersky Automated Security Awareness Platform integration. If necessary, you can also create a secret of the **token** type by specifying the token obtained in the ASAP platform. To do so:

   a. In the **Secret** drop-down list, click **Create**.

      This opens the **Create secret** window.

   b. In the **Name** field, enter the name of the secret. Must contain 1 to 128 Unicode characters.

   c. In the **Token** field, enter the authorization token for API requests to ASAP.

   d. If necessary, select the tags for the secret from the **Tags** drop-down list.

   e. If necessary, add the secret description in the **Description** field.

   f. Click **Create**.

4. In the **ASAP Open API URL** field, specify the address used by ASAP for API requests.
5. In the **ASAP administrator email** field, specify the email address of the ASAP administrator who receives notifications when users are added to the learning groups using KUMA.
6. In the **Notification template** drop-down list, select the template that you want KUMA to use for notifications. If necessary, you can also create a notification template by clicking **Create**.
7. If necessary, in the **Proxy** drop-down list select the proxy server resource to be used to connect to ASAP.
8. Click **Save**.

Integration with ASAP 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 ASAP and changing learning groups

After configuring the integration between ASAP and KUMA, the following information from ASAP 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 ASAP:

1. In the KUMA web interface, in the **Alerts** or **Incidents** section, select the required 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 **ASAP courses details** tab.

The window displays information about the user from ASAP.

You can change the learning group of a user in ASAP.

To change a user learning group in ASAP:

1. In the KUMA web interface, in the **Alerts** or **Incidents** section, select the required 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 ASAP group** drop-down list, select the ASAP learning group you want to assign the user to.

4. Click **Apply**.

   The user is moved to the selected ASAP group, the ASAP 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 ASAP 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.
Compatibility is confirmed only for KUMA 2.0 and later.
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 sending notifications in KUMA

To configure the sending of KUMA notifications to Telegram:

1. Create a response rule:

   a. In the KUMA web interface, 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 a name for 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 **Resources** → **Correlators** section, 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.
Compatibility is confirmed only for KUMA 2.0 or later and UserGate 6.0 or later.
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 web interface, 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:

   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 web interface, 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 a name for 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.
Compatibility is confirmed only for KUMA 2.0 or later and Kaspersky Web Traffic Security 6.0 or later.
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 **Rule 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 web interface, 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. To get the script and the library, please contact Technical Support.
3. Place the script provided by Technical Support 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 web interface, 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 a name for 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.
Compatibility is confirmed only for KUMA 2.0 or later and Kaspersky Secure Mail Gateway 2.0 or later.
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 web interface, 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. To get the script and the library, please contact Technical Support.
3. Place the script provided by Technical Support 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.

Configuring a response rule for integration with KSMG

To configure a response rule:

1. Create a response rule:
   a. In the KUMA web interface, 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 a name for 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, ksmg.py.

f. In the **Script arguments** field, specify:

- `--host` — address of the KSMG server.
- `--username` — name of the user account created in KSMG.

  You can specify the Administrator account.
- `--password` — KSMG user account password.
- `--rule_id` — ID of the rule created in KSMG.
- Specify one of the options depending on the type of the object being blocked:
  - `--email` — specify the field of the KUMA event from which you want to obtain the URL, for example, `{{.SourceUserName}}`.
  - `--ip` — specify the field of the KUMA event from which you want to obtain the IP address, for example, `{{.SourceAddress}}`.
- `--ntlm` — specify this option if the KSMG user was created with NTLM authentication.

Example:

```
--host <address> --username <user> --password <pass> --ntlm --rule_id <id> --email {{.SourceUserName}}
```

g. In the **Conditions** section, add conditions corresponding to the correlation rules that when triggered require blocking the IP address or email address of the message sender in KSMG.

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.

Kaspersky Managed Detection and Response integration

Kaspersky Managed Detection and Response (hereinafter also MDR) is a solution for automatic detection and analysis of security incidents in your infrastructure using telemetry and advanced machine learning technologies.

You can configure the Kaspersky Managed Detection and Response integration in KUMA to automatically import incidents from the MDR Console into KUMA for further processing and analysis.

To configure the integration:

1. Perform the preliminary configuration.
2. Configure the MDR integration.

Preliminary configuration

To perform the preliminary configuration:

1. Provide access to the MDR Console.

   Make sure that the KUMA server on which you want to configure integration has network access to https://mdr.kaspersky.com on port **TCP/443**

2. In KUMA, create a user with the Tenant administrator role.

   Make sure this user has access to the API and can make POST/events requests.

3. In the MDR Console, generate an API token by following these steps:

   a. In the MDR Console window, select the **Settings** menu item.

   b. Select the **API** tab.

   c. In the upper part of the window, click the **Add** button.

   d. Specify the following settings:

   - **Connection name**.

     The connection name can contain Latin letters, numerals, and special characters. This name is used as the user name when creating incidents, comments, and attachments because the access token is not tied to a specific user.

   - **Access rights**.

     Select the access rights that must be granted to perform actions using the HTTP API.

   - **Tenant**.

     Select one or more tenants, if necessary.

     > If there is no value in the **Tenant** drop-down list for incidents created in the MDR Console, specify **Root without tenants** to let the script discover such incidents when connecting.

   e. In the lower part of the section, click the **Create** button. After that, the values of the following parameters are displayed:

   - **JWT token** (refresh_token) is the token that is used to get an access_token that is needed for authorization when working with the API.
   - **Client ID** (client_id) is an ID that must be sent when making API requests.

4. Click the following link to download the kuma_mdr_integration.tar.gz archive:
   https://box.kaspersky.com/f/11a58e42f63e4cef8741/
5. Configure a certificate chain for MDR by following these steps:
   a. In your browser, open the https://mdr.kaspersky.com link.
   b. Click the padlock icon in the address bar next to the URL of the website.
   c. Select **Connection is secure** and click **Certificate is valid**.
   d. Select the **Details** tab.
   e. For each item in the certificate hierarchy (starting with the lowest, ***.mdr.kaspersky.com**, and ending with the root, **DigiCert Global Root G2**), click the **Export** button.
   f. Add the saved certificates to a .PEM file (for example, mdr.pem). The certificates must be placed in order from the lowest to the root.

Initializing integration

To configure KUMA integration with Kaspersky Managed Detection and Response:

1. Prepare the files on the server by following these steps:
   a. Copy the kuma_mdr_integration.tar.gz archive to the KUMA server (for a distributed installation, to the Core server).
   b. Extract the archive to the /opt directory using the following command:
   ```
   sudo tar -xf kuma_mdr_integration.tar.gz -C /opt
   ```
2. Set up the configuration file by following these steps:
   a. Go to the /opt/mdr/conf directory and open the config.yml file for editing.
   b. In the `General settings` section, specify the following settings:
      - Actual path to the directories (by default, /opt/mdr/*)
      - `client_id` (client ID)
   c. In the `Modules settings` → `kuma` section, specify the following settings:
      - `api_url` is the address of the KUMA API.
      - `username` is the name of the user with access to the API that you added earlier.
      - `password` is password of the added user.
      - `tenantId` is ID of the tenant to which incidents will be imported.
   d. In the `Modules settings` → `logging` section, specify the current path to the directory where the script is stored (by default, /opt/mdr/log).

3. Add an MDR authorization token by following these steps:

   a. In the /opt/mdr/conf directory, open the .refresh_token file for editing and paste the MDR API access token that you generated earlier into this file.

   b. Make sure that the file does not end in a newline character `\n` because this may result in an authentication error. To check, you can run the following commands:

   ```
   # Check for a newline character
   wc -l /opt/mdr/conf/.refresh_token
   # If the output is "1 .refresh_token", delete the character
   perl -p -i -e 'chomp if eof' /opt/mdr/conf/.refresh_token
   # Check again
   wc -l /opt/mdr/conf/.refresh_token
   # The expected output is "0 .refresh_token"
   ```

4. Specify when to start collecting incidents:

   a. Go to the /opt/mdr/conf directory and open the .last_check file for editing.

   b. In this file, specify the timestamp (in milliseconds, 13 digits) from which you want to import incidents.

   To check whether the configuration is correct, we recommend specifying a time before the last incident occurred (for example, `1672520400000`).

5. Update the MDR certificate. In the /opt/mdr/conf/ directory, replace the mdr.pem file with a file containing the up-to-date certificate that you created earlier.

6. Run the main.py integration script using the `python3 ./main.py` command:

   - If you see messages about missing required packages when you start the script for the first time, install these packages.

   - If no error messages are displayed after running the script (except for warnings about an invalid certificate), the integration is configured correctly.

   You can find the script log in the /opt/mdr/log/app.log directory.

7. Check the import of incidents.

   Make sure that incidents created in the MDR Console, starting from the time that you specified in the .last_check file, have been imported into KUMA.

8. Run the script in the background by following these steps:

   a. Stop the currently running script (if running).

   b. Run the main.py script again in the background with the following command:

   ```
   nohup python3 /opt/mdr/main.py &
   ```

9. Configure the script to run automatically after a server restart:

```
sudo crontab -e
@reboot sleep 300 && python3 /opt/mdr/main.py &
```

A delay of 5 minutes (`sleep 300`) is necessary to start the kuma-core service before starting the integration script.

Integration of KUMA with Kaspersky Managed Detection and Response is now configured. From now on, incidents from the MDR Console will be automatically imported into KUMA for processing and analysis.

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.
Compatibility is confirmed only for KUMA 2.0 or later and RedCheck 2.6.8 or later.
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 the KUMA web interface in the **Assets** 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:

- Tenant administrator or Tier 2 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 kuma.example.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 utility 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 Tenant administrator or Tier 2 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 with fqdn=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 the KUMA web interface.

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 administrator rights.
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 @<collector_IP>:<collector_port>
   ```

   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 @@<collector IP>:<collector_port>
   ```

   where:

   - <collector_IP> is the collector IP address.
   - <collector_port> is the collector port.
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 the general settings of KUMA.

Logging in to the solution web interface

Only users with an *Active* user account status can log in to the KUMA web interface.

KUMA does not allow creating multiple sessions for a user at the same time. If you try to log in to the KUMA web interface again, for example, from another browser, the earlier session is automatically ended.

We recommend configuring password requirements and rules for logging in to the web interface of the solution in the security policy.

To log in to the solution web interface:

1. Enter the following address in your browser:

   `https://<IP address or FQDN of KUMA Core server>:7220`

   The web interface authorization page will open and prompt you to enter your login and password.

2. Enter the login of your account in the **Login** field.

3. Enter the password for the specified account in the **Password** field.

   Default login credentials:

   - User name: admin
   - Password: mustB3Ch@ng3d!

   After logging in for the first time, change the password of the admin account

4. Click the **Sign in** button.

   If you enter incorrect credentials, KUMA makes you wait for 3 seconds before you can try to log in again. This delay helps prevent an attacker from brute-forcing a password to log in to the KUMA web interface. If you enter incorrect credentials several times in a row, your user account becomes blocked. The maximum number of unsuccessful login attempts and the lockout duration are governed by the security policy.

The main window of the solution web interface is displayed.

In multitenancy mode, a user logging in to the web interface of the solution for the first can see data only for those tenants that were selected for the user account when it was created.

To log out from the web interface of the solution:

In the KUMA web interface, click your user account name in the lower-left corner of the window, and click the **Logout** button in the user account menu.

Configuring the KUMA Core

To configure KUMA Core settings:

1. Go to the **Settings** → **Other** → **Common** section in the KUMA web interface.
2. 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.
3. 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.
4. 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 solution, which means 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 accelerates merging because the solution 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 for the KUMA Core 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.

5. Click **Save**.

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 **Metrics** → <service type> → <service name> → **Enrichment** → **Errors** section of KUMA 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 **Metrics** → <service type> → <service name> → **IO** → **Output Event Loss** section of KUMA 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 KUMA web console, go to the **Settings** → **Other** → **Services monitoring** section.

2. 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.

3. 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 KUMA web console, go to the **Settings** → **Other** → **Services monitoring** section.
2. 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.
3. 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 KUMA services and the flow of events and correlations, KUMA collects and stores a large amount of data. The VictoriaMetrics time series database is used to collect, store and analyze the data.

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
```

In the KUMA Console, in the **Metrics** section, you can find dashboards with information about metrics and visualization of key performance indicators of various KUMA services. The collected metrics are visualized using the Grafana solution. Selecting the **Metrics** section opens the automatically updated Grafana portal deployed as part of KUMA Core installation process.

What host is the KUMA Core installed on?

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
```

The KUMA Core service configures VictoriaMetrics and Grafana automatically, user participation is not required. Graphs in the **Metrics** section appear with a delay of approximately 1.5 minutes. If the **Metrics** section shows "core: <port number>", this means that the metrics were received from the host on which the KUMA 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

IO—metrics related to the service input and output

Metric name	Description
Processing EPS	Number of events processed per second. Starting with KUMA 4.2, the Processing EPS counter takes into account both the auditd event parts and the final (collated) event. For this reason, the **Processing EPS** metric can increase.
Output EPS	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 sending event packets to the destination. Network errors and errors writing to the disk buffer of the destination are displayed separately.
Output Event Loss	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

Metric name	Description
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

Metric name	Description
EPS	The number of events per second sent to be processed. If a filter is present in the configuration of the collector service, events that meet the filter conditions are discarded, and only events that do not meet these conditions are sent for processing.

Aggregation—metrics related to the aggregation rules

Metric name	Description
EPS (events per second)	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

Metric name	Description
Cache RPS	The number of requests per second to the local cache.
Source RPS	The number of requests per second to an enrichment source. Data is collected only for the **dns** and **cybertrace** enrichment source types.
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.

Metric name	Description
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.

Process—metrics related to processes

Metric name	Description
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.

Metric name	Description
Load	Average load.
CPU	CPU load as a percentage.
Memory	RAM usage (RSS) as a percentage.
Disk	Disk space usage as a percentage.

Correlator metrics

IO—metrics related to the service input and output

Metric name	Description
Processing EPS	Number of events processed per second.
Output EPS	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 sending event packets to the destination. Network errors and errors writing to the disk buffer of the destination are displayed separately.
Output Event Loss	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	Number of bytes transmitted into the network per second.

Correlation—metrics related to correlation rules

Metric name	Description
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.

Metric name	Description
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.
Context Tables OPS	Number of queries to the context table in 1 second with the operation indicated.
Context Tables Records	Current number of records in the context table
Context Tables On-Disk Size	Current size of context table on disk.

Enrichment—metrics related to enrichment rules

Metric name	Description
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

Metric name	Description
RPS	The number of times a response rule was activated per second.

Process—metrics related to processes

Metric name	Description
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.

Metric name	Description
Load	Average load.
CPU	CPU load as a percentage.
Memory	RAM usage (RSS) as a percentage.
Disk	Disk space usage as a percentage.

Storage metrics

IO—metrics related to the status of ClickHouse cluster nodes

Metric name	Description
Output EPS	Number of events per second that were inserted into the ClickHouse node or into the disk buffer of the node.
Output Latency	Latency of inserting a batch either into the ClickHouse node or into the disk buffer of the node.
Output Errors	Number of errors per second when inserting a batch into the ClickHouse node or into the disk buffer of the node.
Output Disk Buffer Size	Size of the disk buffer of the ClickHouse node.
Output Event Loss	Number of events that were permanently lost on each ClickHouse node.
Batch size	The size of the batch of events to be inserted into the ClickHouse node.
Insert interval 5min Q1	Duration of the first quartile of the event insertion interval for the last 5 minutes.

ClickHouse / General—metrics related to the general settings of the ClickHouse cluster

Metric name	Description
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 allocated to the ClickHouse process (depends on the specifications of the server and can be expressed, for example, in GB or MB).
Active parts	The number of active parts. Active parts are data (files on disk) that are currently being used to process queries.
Detached parts (count)	The number of detached (disconnected) parts. Detached parts are data that exists on disk, but do not participate in file read and write operations.
Detached parts (size)	Disk space taken up by detached parts. You can specify the maximum size of detached parts in the range from 1% to 90%. The default setting is 1%. If the size of detached parts exceeds the specified maximum value, KUMA assigns the yellow status to the running storage service in the **Active services** section.

ClickHouse / Insert—metrics related to inserting events into a ClickHouse instance

Metric name	Description
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.

Metric name	Description
Delayed Insert QPS	The number of delayed ClickHouse instance insert queries per second sent to the ClickHouse cluster. Requests 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. Requests 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

Metric name	Description
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

Metric name	Description
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

Metric name	Description
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.

Agents

Metric name	Description
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.

Metric name	Description
Load	Average load.
CPU	CPU load as a percentage.

Metric name	Description
Memory	RAM usage (RSS) as a percentage.
Disk	Disk space usage as a percentage.
Disk used (events)	Percentage of /opt filled with events.
Disk used (except events)	Percentage of /opt filled with data other than events.

KUMA Core metrics

Raft—metrics related to reading and updating the state of the KUMA Core

Metric name	Description
Lookup RPS	Number of lookup procedure requests per second sent to the KUMA 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	Number of Raft (SQLite) propose procedure requests per second sent to the KUMA, 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

Metric name	Description
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

Metric name	Description
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

Metric name	Description
RPS	Number of API requests made to the KUMA Core per second.
Latency	Time in milliseconds spent processing a single API request to the KUMA Core. The median value is displayed.
Errors	Number of errors per second while sending API requests to the KUMA Core.

Notification Feed—metrics related to user activity

Metric name	Description
Subscriptions	Number of clients connected to the KUMA 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	Number of errors per second while sending notifications to users.

Schedulers—metrics related to KUMA Core tasks

Metric name	Description
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	Number of errors that occurred per second while performing tasks.
Alerts Queue	Number of alerts in the queue for insertion into the database.

IO—metrics related to the status of KUMA Core nodes

Metric name	Description
Output Disk Buffer Size	The size of each of disk file buffer on the KUMA Core node for the tasks for sending audit events, monitoring, collecting and analyzing data for correlators.

Process—metrics related to processes

Metric name	Description
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.

Metric name	Description
Load	Average load.
CPU	CPU load as a percentage.
Memory	RAM usage (RSS) as a percentage.
Disk	Disk space usage as a percentage.

KUMA agent metrics

IO—metrics related to the service input and output

Metric name	Description
Processing EPS	Number of events processed per second.
Output EPS	Number of events per second sent to the destination.

Metric name	Description
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 sending event packets to the destination. Network errors and errors writing to the disk buffer of the destination are displayed separately.
Output Event Loss	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.

Process—metrics related to processes

Metric name	Description
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.

Metric name	Description
Load	Average load.
CPU	CPU load as a percentage.
Memory	RAM usage (RSS) as a percentage.
Disk	Disk space usage as a percentage.

Event router metrics

IO—metrics related to the service input and output

Metric name	Description
Processing EPS	Number of events processed per second.
Output EPS	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 sending event packets to the destination. Network errors and errors writing to the disk buffer of the destination are displayed separately.
Output Event Loss	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.

Process—metrics related to processes

Metric name	Description
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.

Metric name	Description
Load	Average load.
CPU	CPU load as a percentage.
Memory	RAM usage (RSS) as a percentage.
Disk	Disk space usage as a percentage.

Metrics of the Metrics service

Stats

Metric name	Description
Version	KUMA version.
Total Datapoints	Total number of data points in the database.
Ingestion rate	Database insert rate (data points per second).
Read requests	Database select rate
Available CPU	Available CPU core count
Uptime	Uptime of the service.
Active series	Number of active time series.
Disk space usage	Disk space occupied.
Available memory	Memory available on the server.
Bytes per point	Average data point size (in bytes).

Overview

Metric name	Description
Datapoints ingestion rate	Database insert rate (data points per second).
Requests rate	HTTP request rate by path.
Active timeseries	Number of active time series.
Query duration 0.99 quantile	Query execution time (99th percentile).
Requests error rate	HTTP request error rate.

Resource usage

Metric name	Description
RSS memory % usage	RSS memory usage (percent).
RSS anonymous memory % usage	Anonymous RSS memory usage (percent).
Memory usage	Memory usage (heap, resident, system).
CPU % usage	CPU load (percent).
Open FDs	File descriptor usage.
CPU	CPU core usage.
Goroutines	Number of Go goroutines.
Disk writes/reads	Disk read and write rate (bytes/s).
Threads	Number of threads.
Disk write/read calls	Number of I/O system calls.
TCP connections rate	Rate of TCP connection creation.
TCP connections	Active TCP connections.
Network usage	Network traffic (inbound/outbound).
CPU spent on GC	CPU time spent on garbage collection.
Go scheduling latency	Latency of the Go scheduler.
Slow inserts %	Number of slow inserts (percent).

Troubleshooting

Metric name	Description
Churn rate	Rate of appearance of new time series and disappearance of old time series.
Assisted merges	Forced merges, which are performed synchronously in the data writing thread when background merge processes cannot catch up with the accumulation of LSM tree parts.
Slow queries rate	Rate of slow queries.
Cache miss ratio	Number of cache misses when looking up data points (percent).
Cache usage % by type	Cache usage by type.
Deduplication rate	Data deduplication rate.
Samples dropped for last 1h	Deleted data points over the last hour.
Restarts	Number of service restarts.

Storage

Metric name	Description
Datapoints ingestion rate	Data point insert rate (data points per second).
Disk space usage - datapoints	Disk space occupied by data points.
Storage full ETA	Estimate of when storage will fill up.
Pending data points	Number of data points waiting to be written.
Datapoints	Number of data points by storage type.
LSM parts max by type	Maximum number of LSM tree parts by type.
Rows ignored for last 1h	Number of rows ignored over the last hour.
Active merges	Number of active merges.

Metric name	Description
Concurrent flushes on disk	Concurrent disk writes.
Merge speed	Data merge speed.
Series read per query	Average number of series per query.
Datapoints read per series	Average number of data points per series.
Datapoints read per query	Average number of data points per query.
Datapoints scanned per query	Number of data points scanned per query.

Alerting

Metric name	Description
Config error	Misconfiguration status.
Errors	Error rate according to alerting rules.
Alerting rules	Number of alerting rules.
Alerts fired total	Total number of alerts fired.
Top 10 groups by evaluation duration	Top 10 groups of alerting rules by execution time.
Rules execution rate	Rate of alerting rule execution.
Rules execution errors	Errors during execution of alerting rules.
Missed evaluations	Missed alert rule evaluations.
Top 10 active alerts	Top 10 active alerts by alerting rules.

Tenant metrics

Tenants Overview—metrics related to tenants

Metric name	Description
License EPS	Number of events per second received within the tenant.

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:

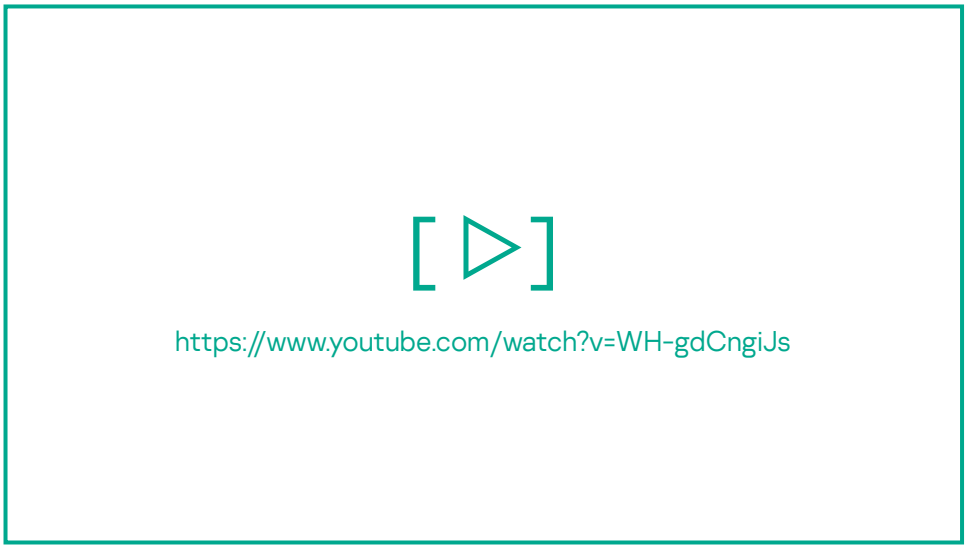
1. Log in to the OS of the server where the KUMA Core is installed.
2. In the file /etc/systemd/system/multi-user.target.wants/kuma-victoria-metrics.service, in the ExecStart parameter, edit the `--retentionPeriod=<metrics storage period, in months>` flag by inserting the necessary period. For example, `--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:

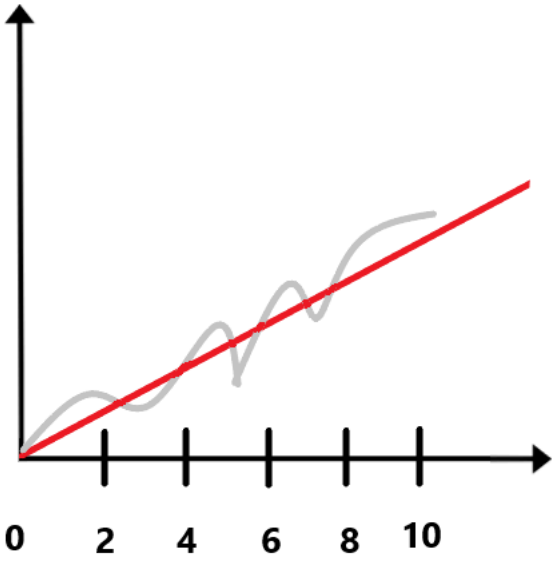
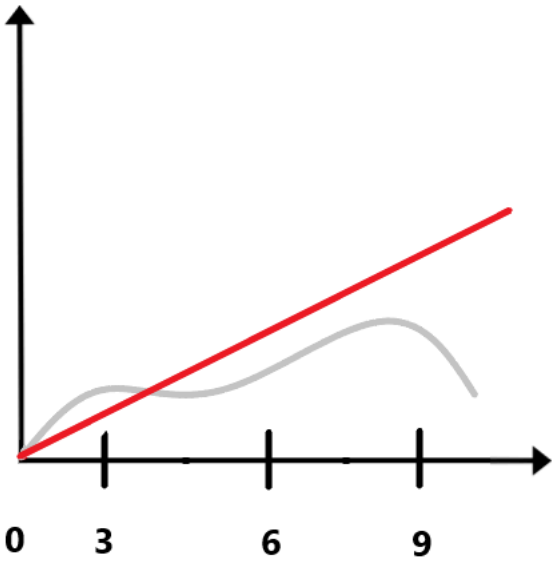
- For a storage service:
  - If an alert was received in response to an API request in the /status parameter from ClickHouse
  - If cold storage of the Storage service is not being monitored
- For a collector service: If an alert was received in response to an API request in the /status parameter.
- 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.
`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.

Error message	Configurable alert parameters	KUMA metric	Description
`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). 
`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. 

Error message	Configurable alert parameters	KUMA metric	Description
`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.

Managing KUMA tasks

When managing the solution web interface, you can use tasks to perform various operations. For example, you can import assets or export KUMA event information to a TSV file.

You can run multiple tasks at the same time. If you run tasks of different types, they are executed in parallel within the same tenant. Tasks of the same type can be run simultaneously only if the tenants are different. If you try to run multiple tasks of the same type in one tenant, only the first such task is run.

Viewing the tasks table

The tasks table contains a list of created tasks and is located in the **Task manager** section of the solution web interface window.

The ability to view other users' tasks depends on the user role. A user with the General administrator role can view the tasks of all users. Users with roles that allow reading and editing shared resources do not have access to tasks. Users with other roles, including users with the Tenant administrator role, can view all tasks in tenants available to them.

By default, the **View just mine** check box is selected in the **Task manager** section. To view all tasks, clear this check box.

The tasks table contains the following information:

- **State**—the state of the task. One of the following statuses can be assigned to a task:
  - **Running**—the task is active.

    You can stop a task of the **Run query in background** type in the **Running** status by clicking the **Stop and start in events** button. Clicking this button and confirming the action opens the **Events** section, where the query is run in normal mode.
  - **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 solution provides the following types of tasks:
  - **Events export** exports KUMA events.
  - **Run query in background** runs an SQL query in the background.
  - **Retrospective scan** scans historical events.
  - **Threat Lookup** requests data from the Kaspersky Threat Intelligence Portal.
  - **KSC assets import** imports asset information from Kaspersky Security Center servers.
  - **KSC assets attributes import** imports information about the attributes of Kaspersky Security Center assets.
  - **KICS/KATA assets import** imports asset information from KICS/KATA.
  - **Accounts import** imports user information from Active Directory.
  - **Run program via KEDR**—starts an arbitrary application using Kaspersky Endpoint Detection and Response.
  - **Repository update** updates the KUMA repository to receive the resource packages from the source specified in settings of the source.
  - **Update monitoring policies** updates the monitoring policy settings.
  - **Clearing resource change history** deletes information about resource changes.
  - **Export event sources** exports event sources in CSV format.
  - **Query to KIRA** requests an analysis of the command using Kaspersky Investigation & Response Assistant.
  - **Partition backup** is the task of backing up partitions according to a schedule or manually.
  - **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**.
- **Created**—the date and time when the task was created.
- **Last updated**—date and time of the last update of the task.
- **Tenant**—the name of the tenant in which the task was started.

The date format depends on a localization language selected in the solution settings. Possible date format options:

- English localization: YYYY-MM-DD.
- Russian localization: DD.MM.YYYY.

Configuring the display of the tasks table

You can customize the display of columns and the order in which they appear in the tasks table.

To customize the display in the task table:

1. In the KUMA web interface, select the **Task manager** section.

   The tasks table is displayed.

2. In the table header, click the gear ( ) icon.

3. In the pane that is displayed:

   - If you want to enable display of a column in the table, select the check box next to the name of the parameter that you want to display in the table.
   - If you do not want the parameter to be displayed in the table, clear the check box.

   At least one check box must be selected.

4. If you want to reset the settings, click the **Default** link.

5. If you want to change the order in which the columns are displayed in the table, move the mouse cursor over the name of the column, hold down the left mouse button and drag the column to the necessary position.

The display of columns in the tasks table will be configured.

Viewing task run results

To view the results of a task:

1. In the KUMA web interface, select the **Task manager** section.

   The tasks table is displayed.

2. Do one of the following:

   - For tasks of the **Events export** and **Export event sources** type, click the link with the task type in the **Task** column and select **Download**.

     Task results are downloaded as a TSV file.

   - For tasks of the **Run query in background** type, click the task type link in the **Task** column. This opens the **Events** section in a new browser tab with the result of the SQL query.

   - Click the name of the user that created the task or **Scheduled task** in the **Created by** column.

     Task results are displayed in the sidebar.

> For **Repository update** tasks, results are not displayed.

The presentation of task results in KUMA depend on the type of the task. For example, for **Threat Lookup**, information about the type of the detected object is displayed along with information about the file, detections, and access paths is displayed. For **KSC assets import** and **KICS/KATA assets import** tasks, information about the tenant and the number of downloaded objects is displayed. For the **Query to KIRA** task, information about task status, the date when the query was sent and the response received, and a link to the event are displayed.

In this section, the **View just mine** check box in the **Created by** column is selected by default. To view all tasks, clear this check box.

Restarting a task

To restart a task:

1. In the KUMA web interface, select the **Task manager** section.

   The tasks table is displayed.

2. Select the check box next to the task you want to restart.

3. Click the **Restart** button.

The task will be restarted.

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 properties.

Only one SMTP server can be added to process KUMA notifications. You can manage the SMTP server connection in the KUMA web interface under **Settings** → **Other** → **General** → **SMTP server settings**.

To configure SMTP server connection:

1. Open the KUMA web interface and select **Settings** → **Other** → **General**.

2. Under **SMTP server settings**, change the relevant settings:

   - Disable the **State** toggle switch 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 you select **Notify once**, the notification about the monitoring policy triggering is sent only once during the whole time when the source has the red status. However, a notification about the monitoring policy triggering is sent every time the source status changes from red to green.

- 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. If, at the time of enabling notifications in KUMA, there are event sources with a red status and you have not yet received notifications about this, you will receive one notification for the whole period, even if the status of the sources changed more than once while monitoring notifications were turned off.

3. 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 web interface.

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

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 Linux installed. The tasks are created in Kaspersky Security Center Web 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 web interface.

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 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 **Assets** section of the KUMA web interface, select the asset that was imported from Kaspersky Security Center.

   The **Asset details** window opens.

2. Click the **KSC response** button.

   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 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 web interface, select the **Resources** section.
2. Select **Correlation rules** and click the **Add correlation rule** button.
3. On the **General** tab, specify 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:

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 KSC.

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 *Anti-Virus scan* task to start when Kaspersky Security Center registers a *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, specify 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 web interface, 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 web interface, 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 KSC**.
6. In the **Kaspersky Security Center 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 workers that the service can run simultaneously to process events in parallel. Specify a positive integer number from 0 to 999. If 0 is specified, the number of logical CPU cores is used.
9. In the **Description** field, you can add up to 4,000 Unicode characters.
10. 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 web interface, 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 web interface, 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 notifications

Notifications in the KUMA web interface

By default, the **Notify about alerts in web interface** check box is selected for all users, and alert notifications are displayed in the KUMA web interface.

If necessary, you can turn off the display of such notifications by clearing the check box in the user profile.

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. Only a user with the General administrator role can receive email notifications.

If the **Receive email notifications** check box is selected for a user with the General administrator role, after enabling the setting, an email notification is sent to the user every 6 hours in accordance with the following rule:

- If at least one server has a non-empty Warning field at the time scheduled for sending the message, the message is sent.
- One message is sent for all services with the yellow status. If no services have the yellow status, no message is sent.

The 6-hour interval is respected unless the KUMA Core is restarted. After each restart of the Core, the 6-hour interval is reset.

KUMA automatically notifies users about the following events:

- A report was created (the users listed in the report template receive a notification).
- An alert was created (all users receive a notification).
- An alert was assigned to a user (the user to whom the alert was assigned receives a notification).

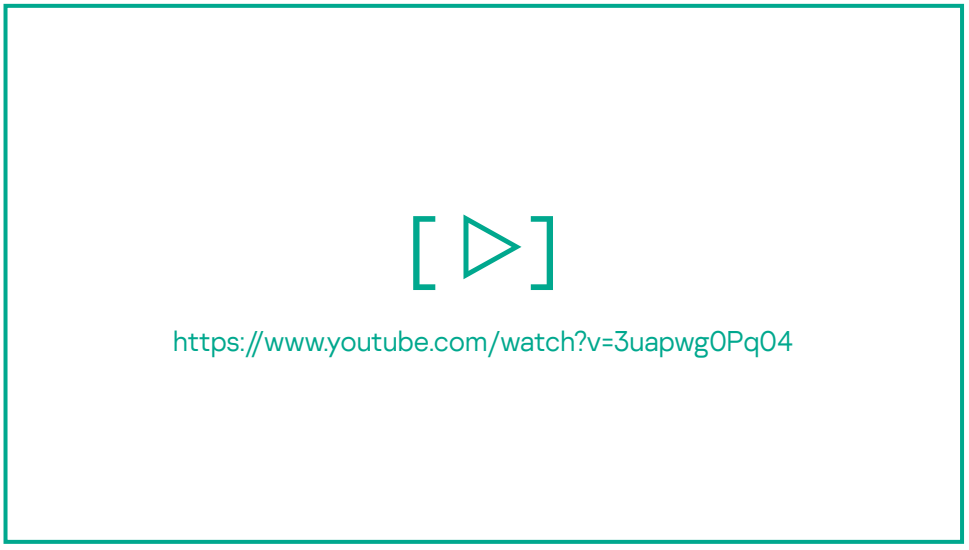
- A task was performed (the users who created the task receive a notification).
- New resource packages are available. They can be obtained by updating the KUMA repository (the users whose email address is specified in the task settings are notified).
- The daily average EPS has exceeded the limit set by the license.
- The hourly average EPS has exceeded the limit set by the SMB license.

Custom notifications

Instead of the standard KUMA notifications about the alert generation, you can send notifications based on custom templates. To configure custom notifications instead of standard notifications, take the following steps:

- Create an email template.
- Create a notification rule that specifies the correlation rules and email addresses.

When an alert is created based on the selected correlation rules, notifications created based on custom email templates will be sent to the specified email addresses. Standard KUMA notifications about the same event will not be sent to the specified addresses.



KUMA logs

KUMA provides the following types of logs:

- Installer logs
- Component logs

You can also generate a report with diagnostic information about your device using the collect.sh utility. For detailed information on using the utility, please refer to the Knowledge Base.

Installer logs

KUMA automatically creates files containing logs of installation, reconfiguration, or removal.

The logs are stored in the ./log/ subdirectory in the installer directory. The name of the log file reflects the date and time when the corresponding script was started.

Names are generated in the following formats:

- Installation log: install-YYYYMMDD-HHMMSS.log. For example: install-20231031-102409.log
- Removal logs: uninstall-YYYYMMDD-HHMMSS.log. For example: uninstall-20231031-134011.log
- Reconfiguration logs: expand-YYYYMMDD-HHMMSS.log. For example: expand-20231031-105805.log

KUMA creates a new log file each time the installation, reconfiguration, or removal script is started. Log rotation or automatic deletion is not performed.

The log incorporates the lines of the inventory file used when the corresponding command was invoked, and the ansible log. For each task, the following information is listed in this order: task start time (Tuesday, October 31, 2023 10:29:14 +0300), run time of the previous task (0:00:02.611), and the total time passed since the installation, reconfiguration, or removal was initiated (0:04:56.906).

Example:

TASK [Add columns to the replicated table] ***************************************

Tuesday, October 31, 2023 10:29:14 +0300 (0:00:02.611) 0:04:56.906 *******

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 Core logs are stored in the /opt/kaspersky/kuma/core/<`KUMA Core service ID`>/log/corecore directory and are archived when they reach the size of 5 GB or the age of 7 days, whichever occurs first. These conditions are checked once daily. 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. If the logs fill up quickly, you must have enough disk space to create a copy of the log file and archive it as part of log rotation.

The component logs are appended until the file 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:

Core	You can enable it in the KUMA web interface, in the **Settings** → **Other** → **General** → **Core settings** → **Debug** section. Storage location: /opt/kaspersky/kuma/core/<`KUMA Core service ID`>/log/core You can download Core logs in the KUMA web interface in the **Resources** → **Active services** section. To do this, select the Core service, click the menu icon (⋮) and select **Download** → **Download log**. If KUMA is installed in a high availability configuration, refer to the *Viewing Core logs in Kubernetes* section below.

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 type, collector/correlator/storage`>/<`service ID`>/log/<`service type, collector/correlator/storage`>. You can download service logs in the KUMA web interface in the **Resources** → **Active services** section. To do this, select a service, click the menu icon (⋮) and select **Download** → **Download 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 computers 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 computers 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 computer 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.

Viewing Core logs in Kubernetes

When Core log files reach 100 MB, they are archived and a new log is written. No more than five files are stored at a time. If there are more than five files when a new log appears, the oldest file is deleted.

On worker nodes, you can view the logs of containers and pods residing on these nodes in the file system of the node.
For example:
/var/log/pods/kuma_core-deployment-<UID>/core/*.log
/var/log/pods/kuma_core-deployment-<UID>/mongodb/*.log

To view the logs of all containers in the Core pod:

```
k0s kubectl logs -l app=core --all-containers -n kuma
```

To view the log of a specific container:

```
k0s kubectl logs -l app = core -c <container_name> -n kuma
```

To enable real-time log viewing, add the -f switch:

```
k0s kubectl logs -f -l app=core --all-containers -n kuma
```

To view the logs of the previous pod that was overwritten by a new one (for example, when recovering from a critical error or after redeployment), add the --previous switch:

```
k0s kubectl logs -l app=core -c core -n kuma --previous
```

To access the logs from other hosts that are not included in the cluster, you need the k0s-kubeconfig.yml file containing the access credentials created during KUMA installation, and the locally installed kubectl cluster management utility.
The cluster controller or traffic balancer specified in the server parameter of the k0s-kubeconfig.yml file must be accessible over the network.

The file path must be exported to a variable:
`export KUBECONFIG=/< file path >/k0s-kubeconfig.yml`

You can use kubeclt to view the logs. For example:

```
kubectl logs -l app=core -c mongodb -n kuma
```

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. Before using archives with CSV geodata files, you must convert these to the format that KUMA supports. The names of CSV files being converted inside archives must match the following templates:

- (GeoIP2|GeoLite2|RU-GeoIP)-(City|Country)-Blocks-IPv(4|6).csv
- (GeoIP2|GeoLite2|RU-GeoIP)-(City|Country)-Locations-%s.csv

Conversion can be done using the script below. Make sure that the CSV files with geodata 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","+
```

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 KUMA web interface, open the **Settings** → **Other** → **General** section.

3. Under **Geographic data**, 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 KUMA web interface, open the **Settings** → **Other** → **General** section.

2. Under **Geographic data**, 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 specified in the `1.0.0.0/24` format in the file imported into KUMA, the range has the `1.0.0.0-1.0.0.255` format 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 use 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`

User guide

This section provides information about managing the KUMA SIEM system.

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. 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.

Resources are contained in the **Resources** section, **Resources** block of KUMA web interface. 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 criteria for selecting individual events from the event stream to be sent to processing.
- **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.
- **Data collection and analysis rules**—resources of this type contain rules that allow scheduling SQL queries with aggregation functions to the storage. Data received from SQL queries is then used for correlation.
- **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.

When you click a resource type, a window opens displaying a table with the available resources of this type. The table contains **the following columns**:

You can configure the set of columns for the table and their order, as well as configure the display of information in some columns:

- You can show or hide columns in the menu that you can open by clicking the gear (⚙) icon in the toolbar in the upper part of the table.
- You can reorder the columns by dragging column headings to the left or right.
- You can configure the display of information in a column by clicking the down arrow (⌄) icon in the heading of the column.
- **Name**—the name of the 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.
- **Type**—the type of the resource. Displayed for all types of resources, except **Aggregation rules**, **Enrichment rules**, **Data collection and analysis rules**, **Filters**, **Active lists**, **Proxies**.
- **Resource path**—address in the resource tree. Displayed in the tree of folders, starting from the tenant in which the resource was created.
- **Tags**—tags assigned to the resource. A resource can have more than one tag.

  Tags are part of the resource and are imported with the resource.

- **Package name**—the name of the package in which the resource was imported from the repository.
- **Correlator—the list of correlators to which the correlation rule is linked**. Displayed only for resources of the **Correlation rule** type.

  For a rule that is not linked to any correlator, the column is blank. If the list contains multiple values, these values are sorted alphabetically. You can filter the values in the **Correlator** column by clicking the heading of the column and selecting the correlation rules that you want displayed in the table: **Without correlator** or **With correlator**.

  For rules of the Shared tenant, the correlators of tenants to which you have access and which you selected in the tenant list are displayed.

Values in the **Correlator** column are not editable. If you want to perform actions on correlators, you need to click the rule in the table to open the properties of the rule, and go to the **Correlators** tab.

- **MITRE techniques**—the MITRE matrix techniques that this correlation rule covers. Displayed only for resources of the **Correlation rule** type. When you hover over a value, the name of the rule is displayed.

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.

The table of resources in the lower part displays the number of resources from tenants that are available to you in the table:

- **Total** is the total amount or the amount with the filter or search applied.
- **Selected** is the number of selected resources.

When filters are applied, the resource selection and the **Selected** value are reset. If the amount of resources changes due to the actions (for example, deletion) undertake by another user, the displayed number of resources changes after you refresh the page, perform an action with the resource, or apply a filter.

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 web interface, click **Duplicate**, then click **Save**.
3. A new resource named "[OOTB]<resource_name> - copy" is displayed in the web interface.
4. 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.

The table of resources in the lower part displays the number of resources from tenants that are available to you in the table:

- **Total** is the total amount or the amount with the filter or search applied.
- **Selected** is the number of selected resources.

When filters are applied, the resource selection and the **Selected** value are reset. If the amount of resources changes due to the actions (for example, deletion) undertake by another user, the displayed number of resources changes after you refresh the page, perform an action with the resource, or apply a filter.

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. Select the relevant folder in the folder structure.
2. Right-click to bring up the context menu and select **Delete**.

   The conformation window appears.
3. Click **OK**.

The folder will be deleted.

> The solution 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 **Resources** → **<resource type>** section, 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.
2. Click the **Add <resource type>** button.

   The window for configuring the selected resource type opens. The available configuration parameters depend on the resource type.
3. Enter a resource name in the **Name** field. It must be unique within the selected tenant.
4. Specify the required parameters (marked with a red asterisk).
5. If necessary, specify the optional parameters (not required).
6. 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 **Resources** → **<resource type>** section, find the required resource in the folder structure.
2. Select the check box near the resource you want to move. You can select multiple resources.

   The ⣿ icon appears near the selected resources. The number of selected resources is displayed in the lower part of the table.
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 **Resources** → **<resource type>** section, find the required resource in the folder structure.
2. Select the check box next to the resource that you want to copy and click **Duplicate**.

   The number of selected resources is displayed in the lower part of the table.

   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 **Resources** → **<resource type>** section, find the required resource in the folder structure.
2. 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. Do one of the following:

   - Click **Save** to save your changes.
   - Click **Save with comment**, and in the displayed window, add a comment that describes your changes. The changes are saved and the comment is added to the created version of the resource.

The resource is updated and a new version is created for it. If this resource is used in a service, restart the service to apply the new version of the resource.

If the current resource is not editable (for example, you cannot edit a correlation rule), you can go to the card of another resource by clicking the **View** button. This button becomes available in batch resources when you click another resource linked to your current resource.

If, when saving changes to a resource, it turns out that the current version of the resource has been modified by another user, you are prompted to select one of the following actions:

- Save your changes as a new version of the resource on top of the changes made by the other user.
- Save your changes as a new resource.

  In this case, a duplicate of the original resource is created with the changed settings. The `- copy` string is added to the name of the new resource, and the name and version of the resource that was duplicated is specified in the version comments of the new resource.
- Discard your changes.

  Discarded changes cannot be restored.

To delete the resource:

1. In the **Resources** → **<resource type>** section, find the required resource in the folder structure.
2. Select the check box next to the resource that you want to delete and click **Delete**.

   The number of selected resources is displayed in the lower part of the table. A confirmation window opens.
3. Click **OK**.

The resource and all its saved versions are deleted.

Bulk deletion of resources

In the KUMA web interface, you can select multiple resources and delete them.

You must have the right to delete resources.

To delete resources:

1. In the **Resources** → **<resource type>** section, find the required resource in the folder structure.
2. Select the check boxes next to the resources that you want to delete.

   In the lower part of the table, the total number of resources and the number of resources selected will be displayed.
3. Click **Delete**.

   This opens a window that tells you whether it is safe to delete resources, depending on whether the resources selected for deletion are linked to other resources.

   For all resources that cannot be deleted, the solution displays a table of links in a modal window.
4. Click **Delete**.

   Only resources without links are deleted.

Deleting folders with resources

You can select the delete operation for any folder at any level, except the tenant.

To delete a folder with resources:

1. In the **Resources** section, select a folder.
2. Click the ⋮ button and select the **Delete** option.

   This opens a window prompting you to confirm deletion. The window displays a field in which you can enter the generated value. Also, if dependent resources exist in the folder, a list of dependencies is displayed.
3. Enter the generated value.
4. Confirm the deletion.

You can delete a folder if:

- The folder does not contain any subfolders or resources.
- The folder does not contain any subfolders, but does contain unlinked resources.
- None of the resources in the folder are dependencies of anything (services, resources, integrations).

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 KUMA web interface, go to the **Resources** → **<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 KUMA web interface, go to the **Resources** → **<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 products 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 **Settings** → **Repository update** section, configure the **Data refresh interval in hours**. The default value is 24 hours.

2. Specify the **Update source**. The following options are available:

   - Kaspersky update servers.

     You can view the list of update 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.

3. If necessary, in the **Proxy server** list, select an existing proxy server to be used when running the **Repository update** task.

   You can also create a new proxy server by clicking the plus (+) icon.

4. 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.

5. 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** → **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.

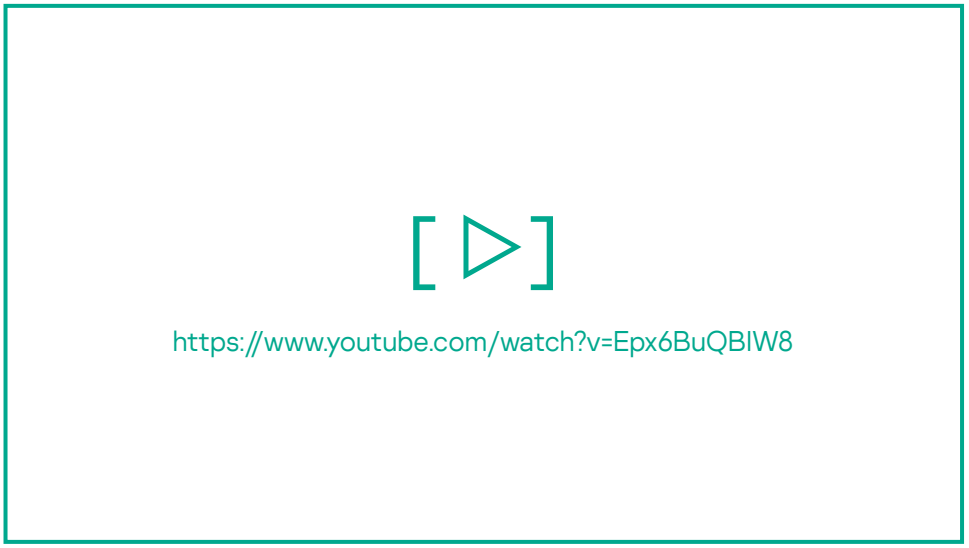
3. If necessary, in the **Proxy server** list, select an existing proxy server to be used when running the **Repository update** task.

   You can also create a new proxy server by clicking the plus (+) icon.

4. 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.

5. Click **Run update**. Thus, you simultaneously save the settings and manually start the **Repository update** task.



https://www.youtube.com/watch?v=Epx6BuQBIW8

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 4.2:
   - Select **Application list**, then click the **Select** button. This opens a window; in that window, in the **Perimeter control** section, select the check box next to KUMA 4.2 to enable updates.
   - 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_4_2=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 KUMA, in the **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 KUMA, in the **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 **Resources** section, click **Export resources**.

   The **Export resources** window opens with the tree of all available resources.

2. In the **Password** field enter the password that must be used to protect exported data.

3. In the **Tenant** drop-down list, select the tenant whose resources you want to export.

4. Check boxes near the resources you want to export.

   If selected resources are linked to other resources, linked resources will be exported, too. The number of selected resources is displayed in the lower part of the table.

5. Click the **Export** button.

Current versions of resources in a password-protected file are saved on your computer in accordance with your browser settings. Previous versions of the resources are saved in the file. The Secret resources are exported blank.

To export a previous version of a resource:

1. In the KUMA web interface, in the **Resources** section, select the type of resources that you need.

   This opens a window opens with a table of available resources of this type.

   If you want to view all resources, in the **Resources** section, go to the **List** tab.

2. Select the check box for the resource whose change history you want to view, and click the **View versions history** button in the upper part of the table.

   This opens the window with the version history of the resource.

3. Click the row of the version that you want to export and click the **Export** button in the lower part of the displayed window.

   You can only export a previous version of a resource. The **Export** button is not displayed when the current version of the resource is selected.

The resource version is saved in a JSON file on your computer in accordance with your browser settings.

Importing resources

In KUMA 4.2, we recommended using resources from the "[OOTB] KUMA 4.2 resources" package and resources published in the repository after the release of this package.

KUMA supports importing resources from an earlier version to a later version, for example from KUMA 3.4 to KUMA 4.0.

To import resources:

1. In the **Resources** section, click **Import resources**.

   The **Import resources**window opens.

2. In the **Tenant** drop-down list, select the tenant that will own the imported resources.

3. 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.

     Imported resources marked as **This resource is a part of the package. You can delete it, but it is impossible to edit.** can only be deleted. To rename, edit or move an imported resource, make a copy of the resource using the **Duplicate** button and perform the desired actions with the copy of the resource. When importing future versions of the package, the duplicate is not updated because it is a separate object.

     You can edit Imported resources in the `Integration` folder. Such resources are marked as **This resource is a part of the package** To a package resource in the `Integration` folder, you can add a dictionary of the `Table type`; adding other resources is not allowed. When importing subsequent versions of the package, the edited resource will not be replaced with the corresponding resource from the package, which allows you to keep the changes you made.

4. 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 make sure the imported resources work as intended, you need to create the corresponding extended schema fields in the **Settings** → **Extended extended event schema fields** section or import a normalizer that uses the required fields.

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.

If a normalizer is imported into KUMA that uses an extended event schema field that does not meet the KUMA requirements, the import is completed, but the extended event schema field is created with the **Disabled** status and you cannot use this field in other normalizers and resources. An extended event schema field runs afoul of requirements if, for example, its name contains special characters or spaces. If you want to use such a field that does not meet the requirements, you need to fix its problems (for example, by renaming it) and then enable the field.

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.

In this case, if a conflict occurs, the imported resource is added as a new version of the existing resource. A *resources imported* comment is added to this version.

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 object** 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 linked 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 *Import conflicting resources in the same tenant is not allowed* error 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 general administrator can import categories into the Shared tenant.

   The *Only general admin is allowed to import categories into shared tenant* error means that the imported package contains resources with linked shared asset categories. The categories or resources with linked shared asset categories are displayed 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 General administrator account.

6. Only the general administrator can import resources into the Shared tenant.

   The *Only general admin is allowed to import resources into shared tenant* error means that the imported package contains resources with linked shared resources. The resources with linked shared resources are displayed 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 General administrator account.

Resource search

You can search resources by name or tags. You can also use full-text search to search for resources by all of their fields. For resources of the **Correlation rules** type, you can use full-text search to search for correlators in which the rules are used. Searching by filter finds all resources that use the filter.

The search is carried out only on the latest version of the resources.

To find resources:

1. In the KUMA web interface, in the **Resources** section, select the type of resources that you need.

   This opens a window opens with a table of available resources of this type.

   If you want to view all resources, in the **Resources** section, go to the **List** tab.

2. If necessary, to toggle the search mode, click the table icon in the upper part of the resource table.

   You can search by name, tags, or full-text search in all fields of resources. By default, search by name, tags, and correlators is enabled (only for correlation rules).

   You can tell which search mode is currently enabled by the default text that is displayed in the search field.

3. In the **Search** field, start typing the search text.

   The search is initiated as you type characters in the field and is case-insensitive. The table displays only those resources that satisfy the search conditions, and the number of such resources is displayed in the lower part of the table.

   For full-text search, the results are sorted in the descending order of the number of words from the search string found in the fields of a resource. KUMA searches the JSON of a resource; if another resource is specified in it, it searches the specified resource too. If the resource refers to other resources, KUMA also traverses these resources and searches in their content.

4. If you want to reset the search result, clear the **Search** field or click the × icon.

Tag management

To help manage resources, the KUMA web interface 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** → **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. Go to the **Resources** section and 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. Go to the **Resources** section and 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. Go to the **Resources** section.
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.

Resource usage tracing

For stable operation of KUMA, it is important to understand how some resources affect the performance of other resources, what connections exist between resources and other KUMA objects. You can visualize these interdependencies on an interactive graph in the KUMA web interface.

Displaying the links of a resource on a graph

To display the relations of the selected resource:

1. In the KUMA web console, in the **Resources** section, select a resource type.

   A list of resources of the selected type is displayed.
2. Select the resource that you need.
3. In the toolbar or in the menu (⋮), click the **Show dependencies** button.
4. Click the **Show dependencies** button.

   This opens the a window with the dependency graph of the selected resource. If you do not have rights to view the resource, it is marked in the graph with the (inaccessible resource) icon. If necessary, you can close the graph window to go back to the list of resources.

Resource dependency graph

The graph displays all relations that are formed based on the universal unique identifier (UUID) of resources used in the configuration of the resource selected for display, as well as relations of resources that have the UUID of the selected resource in their configuration. Downward links, that is, resources referenced (used) by the selected resource, are displayed down to the last level, while for upward links, that is, resources that reference the selected resource, only one level is displayed.

On the graph, you can view the dependencies of the following resources:

- Correlation rules
- Aggregation rules
- Enrichment rules
- Response rules
- Data mining rules
- Normalizers
- Connectors
- Destinations
- Filters
- Dashboards

  A peculiarity of link display: dashboards do not have an "upward" link because dashboards are not part of other resources.

- Report templates

  A peculiarity of link display: report templates do not have an "upward" link because report templates are not part of other resources.

- Notification templates
- Active lists
- Dictionaries
- Proxy servers
- Secrets
- Context tables

- Collectors

  Note that if a collector was initially selected for displaying links, "upward" links are not displayed.

- Correlators

- Storages

- Agents (autoagents)

  Note that if an agent is selected for displaying links, the collector is displayed with the **linked** relation type only if the collector is running as a service and in the collector is correctly (fqdn+port) specified as the destination of the agent.

- Event routers

  Note that if an event router was initially selected for displaying links, "upward" links are not displayed.

- Integrations

  The name of the integration corresponds to the name of the tab in the **Integrations** section.

- 

Resource group

  The number before the parentheses indicates the number of resources from the group displayed in the graph; the number in parentheses indicates the total number of resources in the group.

- Inaccessible resource (if you do not have the rights to view it).

Clicking a resource node lets you view the following information about the resource:

- **Name**

  Contains a link to the resource; clicking the link opens the resource in a separate tab; this does not close the graph window.

- **Type**

- **Path**

  Resource path without a link.

- **Tags**

- **Tenant**

- **Package name**

You can open the context menu of the resource and perform the following actions:

- Show relations of resource

  The dependencies of the selected resource are displayed.

- Hide resource on graph

  The selected resource is hidden. Resources at the lower level that the selected resource references are marked with "*" as having hidden links. Resources that refer to a hidden resource are marked with the link ( ) icon as having hidden links. In this case, the graph becomes broken.

- Hide "downward" relations of resource on graph

  The selected resource remains. Only those lower level resources that do not have any links remaining on the first higher level on the graph are hidden. Resources referenced by resources of the first (hidden) level are marked with "*" as having hidden links.

- Hide all resources of this type on graph

  All resources of the selected type are hidden. This operation is applied to each resource of the selected type.

- Update resource relations

  You can update the resource state if the resource was edited by another user while you were managing the graph. Only changes of visible links are displayed.

- Group

  If there is no group node on the screen: the group node appears on the screen, resources of the same type as the selected resource and resources that refer to the same resource are hidden. The edges are redrawn from the group. The **Group** button is available only when more than 10 links to resource of the same type exist.

  If there is a group node on the screen: the resource is hidden and added to the group, the edges are redrawn from the group.

Several types of relations are displayed on the graph:

- Solid line without a caption.

  Represents a direct link by UUID, including the use of secrets and proxies in integrations.

- Line captioned **<function_name>**.

  Represents using an active list in a correlation rule.

- Dotted line captioned **linked**.

  Represents a link by URL, for example, of a destination with a collector, or of a destination with a storage.

Resources created inline are shown on the graph as a dotted line with the **linked** type.

We do not recommend building large dependency graphs. We recommend limiting the number of nodes to 100 nodes.

When you open the graph, the resource selected for display is highlighted with a blinking circle for some time to set it apart graphically from other resources and draw attention to it.

You can look at the map of the graph to get an idea of where you are on the graph. You can use the selector and move it to display the necessary part of the graph.

By clicking the **Arrange** button, you can improve the display of resources on the graph.

If you select **Show links**, the focus on the graph does not change, and the resources are displayed so that you do not have to return to where you started.

When you select a group node in the graph, a sidebar is displayed, in which you can hide or show the resources that are part of the group. To do so, select the check box next to the relevant resource and click the **Show on graph** or **Hide on graph** button.

The graph retains its state if you displayed something on the graph, then switched to editing a resource, and then reopened the graph tab.

The previously displayed resources on the graph remain in their places when new resources are added to the graph.

When you close the graph, all changes are discarded.

After the resource links are drawn on the graph, you can search for a node:

- By name
- By tag
- By path
- By package

Nodes, including groups that match the selection criterion, are highlighted with a yellow circle.

You can filter the graph by resource type:

- Hide or show resources of a certain type.
- Hide resources of multiple types. Display all types of resources.

With the filter window closed, you can tell the selected filters by the indicator, a red dot in the toolbar.

Your actions when managing the graph (the last 50 actions) are saved in memory; you can undo changes by pressing **`Ctrl/Command+Z`**.

You can save the displayed graph can be saved to an SVG file. The visible part of the graph is saved in the file.

Resource versioning

KUMA stores the change history of resources in the form of versions. A resource version is created automatically when you create a new resource or save changes made to the settings of an existing resource.

> The change history is not available for the **Dictionaries** resource. To save the history of dictionaries, you can export data.

Resource versions are retained for the duration specified in the **Settings** section. When the age of a resource version reaches the specified value, the version is automatically deleted.

You can view the change history of KUMA resources, compare versions, and restore a previous version of a resource, for example, if it fails and you need to recover it.

To view the change history of a resource:

1. In the KUMA web interface, in the **Resources** section, select the type of resources that you need.

This opens a window opens with a table of available resources of this type.

If you want to view all resources, in the **Resources** section, go to the **List** tab.

2. Select the check box for the resource whose change history you want to view, and click the **View versions history** button in the upper part of the table.

This opens a window with a table of saved versions of the selected resource. New resources have only one version, the current version.

For each version, the table displays the following information:

- **Version** is the serial number of the resource version. When you save changes to the resource and create a new version, the serial number is increased by 1.

  The version with the highest number and the most recent publication date reflects the current state of the resource. Version 1 reflects the state of the resource at the moment when it was created.

- **Published** is the date and time when the resource version was created.

- **Author** is the login of the user that saved the changes to the resource.

  If the changes were made by the system or by the migration script, the displayed value is **system**.

- **Comment** is a text comment added by the author when saving changes, or a system comment describing the changes made.

- **Retention period** is the number of days and the date after which the resource version will be deleted.

  If necessary, you can configure the retention period for resource versions.

- **Actions** is the button that restores the resource version.

You can sort the table of resource versions by the **Version**, **Published**, and **Author** columns by clicking the header and selecting **Ascending** or **Descending**. You can also display only changes made by a specific author or authors in the table by clicking the heading of the **Author** column and selecting the authors as needed.

If you want to view the status of a resource in a specific version, click that version in the table. This opens a window with the resource of the selected version, in which you can:

- View the settings specified in that version of the resource.

- Restore this version of the resource by clicking the **Restore** button.

- Export this version of the resource to a JSON file by clicking the **Export** button.

Comparing resource versions

You can compare any two versions of a resource, for example, if you need to track changes.

To compare versions of a resource:

1. In the KUMA web interface, in the **Resources** section, select the type of resources that you need.

   This opens a window opens with a table of available resources of this type.

   If you want to view all resources, in the **Resources** section, go to the **List** tab.

2. Select the check box next to a resource and click the **View versions history** button in the upper part of the table.

   This opens the window with the version history of the resource.

3. Select the check boxes next to the two versions of the resource that you want to compare and click the **Compare** button in the upper part of the table.

This opens the resource version comparison window. Resource fields are displayed as a list or in JSON format. Differences between the two versions are highlighted. You can select other versions to compare using the drop-down lists above the resource fields.

Restoring a resource version

You can restore a previous version of a resource, for example, if you need to recover the resource in case of mistakes made when making changes.

> Versions of automatically generated agents cannot be restored separately because they are created when the parent collector is modified. If you want to restore a version of an automatically generated agent, you need to restore the corresponding version of the parent collector.

To restore a previous version of a resource:

1. In the KUMA web interface, in the **Resources** section, select the type of resources that you need.

   This opens a window opens with a table of available resources of this type.

   If you want to view all resources, in the **Resources** section, go to the **List** tab.

2. Select the check box next to a resource and click the **View versions history** button in the upper part of the table.

   This opens the window with the version history of the resource.

3. In the row of the relevant version, in the **Action** column, click the **Restore** button.

   You can also restore a version by clicking the row of this version and clicking the **Restore** button in the lower part of the window.

   > You can restore only previous versions of a resource; for the current version, the **Restore** button is not available.

   > If the structure of the resource has changed after a KUMA update, restoring its saved versions may not be possible.

4. Confirm the action and, if necessary, add a comment. If you do not add a comment, the *Restored from v. <number of the restored version>* comment is automatically added to the version.

The resource version is restored as a new version and become the current version.

If the resource for which you restored the version is added to the active service, this also changes the state of the service. You must restart the service to apply the resource change.

Configuring the retention period for resource versions

You can change the retention period of resource versions in the KUMA web interface in the **Settings** → **Other** → **General** section by changing the **Resource history retention period, days** setting.

The default setting is 30 days. If you want to keep all versions of resources without time limits, specify 0 (store indefinitely).

> Only a user with the General administrator role can view and manage the retention period of resource versions.

The retention period of resource versions is checked daily, and versions of resources that have been stored in KUMA for longer than the specified period are automatically deleted. In the **Task manager** section, the *Clearing resource change history* task is created to check the storage duration of resource versions and delete old versions. This task also runs after a restart of the Core component.

You can check the time remaining until a resource version is deleted in the table of versions, in the **Retention period** column.

Destinations

Destinations define network settings for sending normalized events. Collectors and correlators use destinations to describe where to send processed events. Typically, correlators and storages act as destinations.

You can specify destination point settings on the **Basic settings** and **Advanced settings** tabs. The available settings depend on the selected type of destination.

Destinations can have the following types:

- internal for receiving data from KUMA services using the 'internal' protocol.
- nats-jetstream for communication through NATS.
- tcp for communications over TCP.
- udp for communication over UDP.
- http for communication over HTTP.
- diode for transmitting events using a data diode.
- kafka for kafka communications.
- file for writing to a file.
- storage for sending data to storage.
- correlator for sending data to a correlator.
- eventRouter for sending events to an event router.

Destination, internal type

Destinations of the **internal** type are used for receiving data from KUMA services using the 'internal' protocol. You can transfer the following data using 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 type, at the **Event parsing** step, you can pass the name of the file being processed by the collector or the path to the file in the KUMA event field in the **Mapping** table. 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** 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. 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.
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: 16777216 (16 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.
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.
Compression	Determines whether the stream of events is sent by the KUMA agent to the destination in compressed form. Available values: • **Disabled**. • **Snappy** • **gzip**
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.
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	The 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?** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

Setting	Description
	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 services, that is, 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, type nats-jetstream

Destinations of the **nats-jetstream** type are used for communication through NATS. Settings for a destination of the **nats-jetstream** 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: **nats-jetstream**. 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. 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
Subject	The topic of NATS messages. Characters are entered in Unicode encoding. Required setting.
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.
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.
	• **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.
	• **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 displayed in the **PFX secret** drop-down list.
	The **Custom PFX** mode is not displayed when **Authorization** is set to **Plain**.

Compression	Determines whether the stream of events is sent by the KUMA agent to the destination in compressed form. Available values: • **Disabled**. • **Snappy** • **gzip**.
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.

Setting	Description
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	The 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?** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

Setting	Description
	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 services, that is, 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, tcp type

Destinations of the **tcp** type are used for TCP communications. Settings for a destination of the **tcp** 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: **tcp**. 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. 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
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**
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.
Compression	Determines whether the stream of events is sent by the KUMA agent to the destination in compressed form. Available values: • **Disabled**. • **Snappy** • **gzip**
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	The 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?** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

Setting	Description
	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 services, that is, 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, 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. 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
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**
Compression	Determines whether the stream of events is sent by the KUMA agent to the destination in compressed form. Available values: • **Disabled**. • **Snappy** • **gzip**
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.
Debug	The 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?** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

Setting	Description
	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 services, that is, 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, http type

Destinations of the **http** type are used for HTTP communications. Settings for a destination of the **http** 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: **http**. 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. 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.
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.
	• **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.
	• **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 displayed in the **PFX secret** drop-down list.
	The **Custom PFX** mode is not displayed when **Authorization** is set to **Plain**.

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.
Compression	Determines whether the stream of events is sent by the KUMA agent to the destination in compressed form. Available values: • **Disabled**. • **Snappy** • **gzip**

Setting	Description
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.
Path	The path that must be added in the request to the URL specified in the **URL** field on the **Basic settings** tab. For example, if you specify `/input` as the path and enter `10.10.10.10` for the URL, the destination will make requests to `10.10.10.10/input`.
Health check timeout	Interval, in seconds, for checking the health of the destination.
Health check path	The URL for sending requests to obtain health information about the system that the destination resource is connecting to.
Health check	This toggle switch enables the health check. This toggle switch is turned off by default.
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	The 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?** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

Setting	Description
	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 services, that is, 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, diode type

Destinations of the **diode** type are used to transmit events using a data diode. Settings for a destination of the **diode** 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: **diode**. Required setting.

Setting	Description
Data diode source directory	Path to the directory from which the data diode moves events. The maximum length of the path is 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\\KUMA` These patterns are used by the system to validate the path. The double backslashes (\\) are used as part of the regular expression syntax and must not be manually specified in the path itself. **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/` 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.

Setting	Description
	Files are available at the following paths: • `/opt/kaspersky/kuma/clickhouse/logs/` • `/opt/kaspersky/kuma/metrics/log/` The paths specified in the **Data diode source directory** and **Temporary directory** fields may not be the same.
Temporary directory	Path to the directory in which events are prepared for transmission to the data diode. The maximum length of the path is 255 Unicode characters. Events are stored in a file when a timeout or a buffer overflow occurs. The default timeout is 10 seconds. The prepared file with events is moved to the directory specified in the **Data diode source directory** field. The checksum (SHA-256) of the file contents is used as the name of the file with events. The paths specified in the **Data diode source directory** and **Temporary directory** fields may not be the same.
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.
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.
Compression	Determines whether the stream of events is sent by the KUMA agent to the destination in compressed form. Available values: • **Disabled**. • **Snappy** • **gzip**.
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.
Debug	The 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?** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

Setting	Description
	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 services, that is, 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, kafka type

Destinations of the **kafka** type are used for communication using kafka. Settings for a destination of the **kafka** 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: **kafka**. 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. 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
Topic	Subject of Kafka messages. The maximum length of the subject is 255 characters. You can use the following characters: a–z, A–Z, 0–9, ".", "_", "-". Required setting.
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 PFX certificate with a private key in PKCS#12 container format in an external Certificate Authority, export the PFX certificate from the key store, and 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.
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.

Setting	Description
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**

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.

Compression	Determines whether the stream of events is sent by the KUMA agent to the destination in compressed form. Available values: • **Disabled**. • **Snappy** • **gzip** • **LZ4** • **zstd** If you do not select a value, compression is disabled (default 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.
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	The 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?** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

Setting	Description
	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 services, that is, 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, file type

Destinations of the **file** type destinations are used for writing to a file. Settings for a destination of the **file** type are described in the following tables.

When deleting a destination of the **file** type that is being used in a service, you must restart the service.

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: **file**. Required setting.

Setting	Description
URL	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/` 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/` 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**
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.
Debug	The 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?** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

Setting	Description
	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 services, that is, 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, storage type

Destinations of the **storage** type are used to transfer data to storage. Settings for a destination of the **storage** 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: **storage**. 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. 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
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.
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.
Compression	Determines whether the stream of events is sent by the KUMA agent to the destination in compressed form. Available values: • **Disabled**. • **Snappy** • **gzip**
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.
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	The 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?** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

Setting	Description
	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 services, that is, 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, correlator type

Destinations of the **correlator** type are used to transfer data to a correlator. Settings for a destination of the **correlator** 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: **correlator**. 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. 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
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.
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.
Compression	Determines whether the stream of events is sent by the KUMA agent to the destination in compressed form. Available values: • **Disabled**. • **Snappy** • **gzip**
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.
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	The 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?** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

Setting	Description
	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 services, that is, 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** type are used for sending events to the 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. 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
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.
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.
Compression	Determines whether the stream of events is sent by the KUMA agent to the destination in compressed form. Available values: • **Disabled**. • **Snappy** • **gzip**
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.
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	The 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?** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

Setting	Description
	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 services, that is, 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 KUMA 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 the IP address.

  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 KUMA web interface:

   - In the **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 using 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 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 over the additional normalizer, a plus button appears. You can use this button to create a new additional event parsing rule. To delete a normalizer, use the button with the trash icon.

   Extra normalizers are applied from top to bottom until the first condition matches.

   Conditions for processing an event by extra normalizers

If the event matches the conditions to be processed by the extra normalizer, the following may happen:

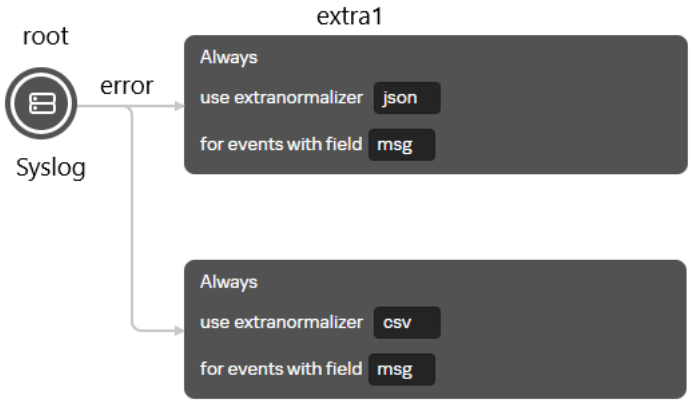
- If the event is processed by an extra normalizer:
  1. After processing, the event is sent to be processed to child extra normalizers, if any are configured.
  2. The processed event is returned to the main normalizer and passed down the chain.
- If the event is not processed, an error occurs, and **Keep raw event** is set to **Only errors** for the collector:

1. The event is returned to the main normalizer.
2. KUMA logs an error in the collector log.
3. The raw event is written to the **Raw** field.

Example 1

Prerequisite:

A collector has been created with a normalizer that has the **Keep original event** setting set to **Only errors**.



Example of error occurrence

Result

The event goes to the `extra 1 JSON` normalizer only if the event matches the selection condition.

The `extra1 JSON` extra normalizer fails to process the event and an error occurs (see the figure above). Processing ends, an error is logged in the collector log, and the raw event is written to the **Raw** field.

Example 2

Prerequisite:

A collector has been created with a normalizer that has the **Keep original event** setting set to **Only errors**.



Diagram of event normalization

Sent event:

```
<1>May 31 11:55:21 srv-06 for_secondlvl_grp1[786]: Message=|test1|test2|;
```



Diagram of error occurrence

Result

An error occurred at the second level of `json1` extra normalization, processing is terminated (see the figure above).

An error is logged in the collector log, and the raw event is written to the **Raw** field.

Example 3

Prerequisite:

A collector has been created with a normalizer that has the **Keep original event** setting set to **Only errors**.

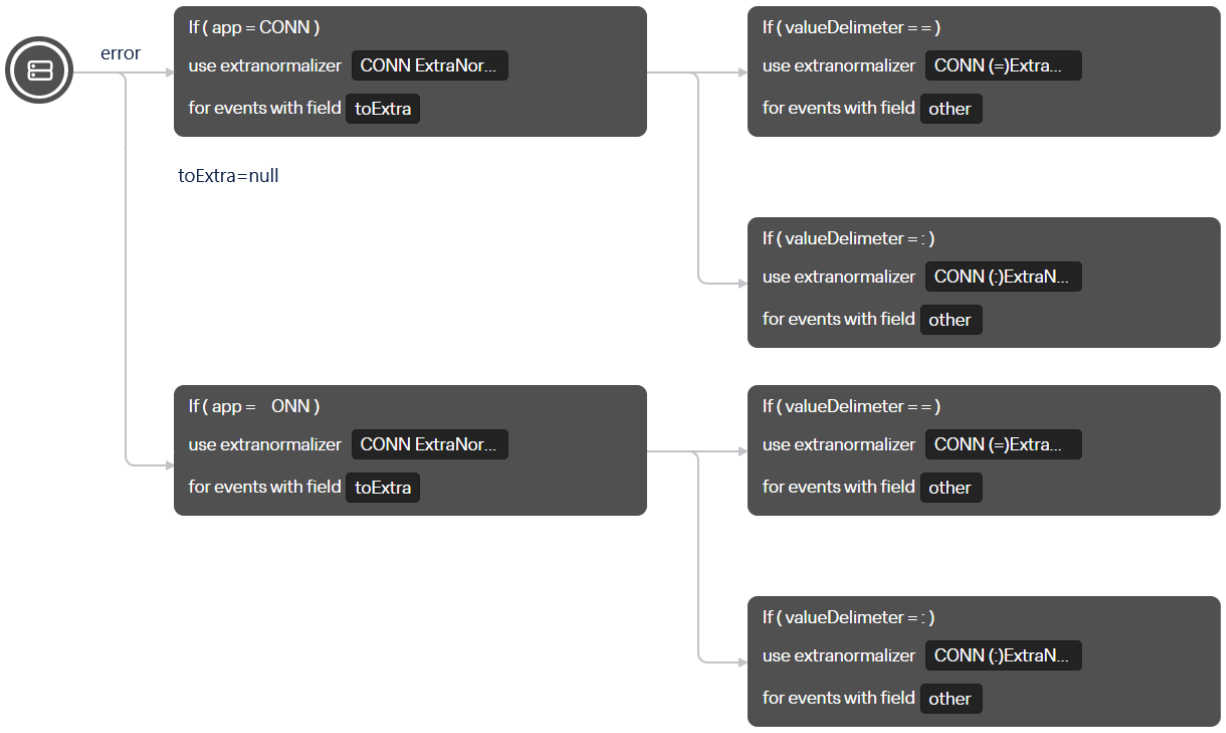


Diagram of error occurrence

Result

The level 1 extra normalizer fails to process the event and an error occurs (see the figure above). The branch of this extra normalizer is terminated and level 2 child extra normalizers are not applied.

Event processing fails, the event is returned to the main normalizer, no error is logged in the collector log, and the raw event is not written to the **Raw** field.

- If the event is not processed, the field passed to the extra normalizer is empty, and **Keep raw event** is set to **Only errors** for the collector:

  1. The event is returned to the main normalizer.
  2. No error is logged in the collector log.
  3. The raw event is not written to the **Raw** field.

  Example 4

  Prerequisite:

A collector has been created with a normalizer that has the **Keep original event** setting set to **Only errors**.

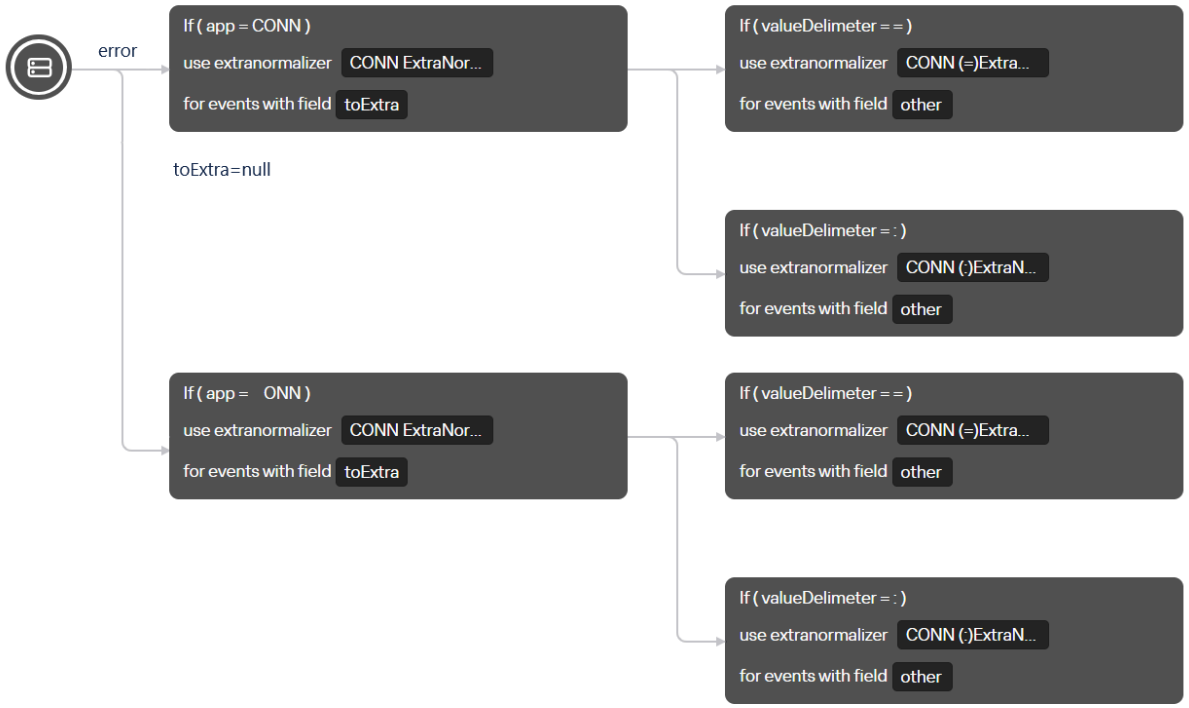


Diagram of error occurrence

Result

The field passed to extra normalizer 1 is empty (see the figure above).

Event processing fails, the event is returned to the main normalizer, no error is logged in the collector log, and the raw event is not written to the **Raw** field.

- If the event does not match the conditions of any extra normalizer, and **Keep raw event** is set to **Only errors** for the collector:

  1. The raw event is written to the **Raw** field.

  2. No error is logged in the collector log.

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 **Resources** → **Normalizers** in the web interface.



https://www.youtube.com/watch?v=wtiMStLixrI

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. Available event parsing settings are listed in the table below.

When normalizing events, you can use extended event schema fields in addition to standard KUMA event schema fields.

Available event parsing settings

Setting	Description

Parsing method

The type of incoming events. Depending on the selected parsing method, you can use the predefined event field matching rules or define your own rules. When you select some parsing methods, additional settings may become available they you must specify. 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, in the **Use raw event** drop-down list, select **Yes**.

- **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 regular expression with named capture groups following the RE2 syntax, 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, in the **Use raw event** drop-down list, select **Yes**.

  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 **Use 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 **Use 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.

- **netflow**

  This parsing method is used to process data in all supported NetFlow protocol formats: NetFlow v5, NetFlow v9, and IPFIX.

  If you select this parsing method, you can use the predefined rules for converting events to the KUMA format by clicking **Apply default mapping**. This takes into account the source fields of all NetFlow versions (NetFlow v5, NetFlow v9, and IPFIX).

  If the **netflow** parsing method is selected for the main parsing, extra normalization is not available.

  > The default mapping rules for the **netflow** 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.

- **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** The normalizer uses this parsing method to process data obtained by making a selection from the database. • **normalizer** This parsing method allows you to reuse a previously created normalizer as the primary or extra normalizer for processing events. If you select this parsing method, in the **Normalizer** drop-down list, you need to select one of the previously created normalizers of the json, cef, regexp, syslog, csv, kv, xml, or sql type. Normalizers with the **normalizer** parsing method cannot be selected. Required setting.
Name	Name of the parsing rule. Maximum length of the name: 128 Unicode characters. The name of the main parsing rule is used as the name of the normalizer. Required setting.
Tenant	The name of the tenant that owns the resource. This setting is not available for extra parsing rules.
Keep raw event	Keeping raw events 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 useful for debugging because an event having a non-empty `Raw` field indicates 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**. • **Always**—always save the raw event in the `Raw` field of the normalized event. Required setting. This setting is not available for extra parsing rules.

Keep extra fields	Keep fields and values for which no mapping rules are configured. This data is saved as an array 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, no extra fields are saved.
	Required setting. This setting is not available for the **normalizer** parsing method.

Description	Description of the resource. Maximum length of the description: 4000 Unicode characters. This setting is not available for extra parsing rules.
Event examples	Example of data that you want to process. This setting is not available for the following parsing methods: **netflow5**, **netflow9**, **sflow5**, **ipfix**, **sql**, and **normalizer**. If the event was parsed successfully, and the type of the data obtained from the raw event matches the type of the KUMA field, the **Event examples** field is filled with data obtained from the raw event. For example, the `"192.168.0.1"` value in quotation marks does not appear in the `SourceAddress` field. However, the `192.168.0.1` value is displayed in the **Event examples** field.

Mapping

Settings for configuring the mapping of source event fields to fields of the event in the KUMA format:

- **Source** lists the names of the raw event fields that you want to convert into KUMA event fields.
  Next to field names in the **Source** column, 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 for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, 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.
  - Converting the IP address to the IPv4 format:
    - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.
    - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

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**.

- **KUMA field** lists fields of KUMA events. You can search for fields by entering their names.
- **Label** is a unique custom label for event fields that begin with `DeviceCustom*` and `Flex*`.

You can add new table rows or delete table rows. To add a new table row, click **Add row**. To delete a single row in the table, click ✕ next to it. To delete all table rows, click **Clear all**.

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.

These settings are not available for the **normalizer** parsing method.

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, in SQL queries for filtering and searching events, in the table of events, or in presets. You can view the list of all extended event schema fields that exist in KUMA in the **Settings** → **Extended event schema fields** 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	Availability in the normalizer	Data type
`S`	All types	String.
`N`	All types	Number.
`F`	All types	Floating point number.
`SA`	KV, JSON	Array of strings. The order of the array elements is the same as the order of the elements of the raw event.
`NA`	KV, JSON	Array of integers. The order of the array elements is the same as the order of the elements of the raw event.
`FA`	KV, JSON	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.

- **Updated at**—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.

- `KL_AI_DLLHijackingCheckResult` , type `N` for storing the check result of a dynamic link library that can potentially be used in DLL Hijacking attacks. The result of the check is provided by the KSN service.

- 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 KUMA web console, in the **Settings** → **Extended event schema fields** section, click the **Add** button in the upper part of the table.

   This opens the **Create extended field** pane.

2. 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.

3. In the **Type** field, select the data type of the extended event schema field.

   Possible data types

Type	Availability in the normalizer	Data type
`S`	All types	String.
`N`	All types	Number.
`F`	All types	Floating point number.
`SA`	KV, JSON	Array of strings. The order of the array elements is the same as the order of the elements of the raw event.
`NA`	KV, JSON	Array of integers. The order of the array elements is the same as the order of the elements of the raw event.
`FA`	KV, JSON	Array of floats. The order of the array elements is the same as the order of the elements of the raw event.

4. 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.

5. 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.

6. 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 KUMA web console, in the **Settings** → **Extended event schema fields** section, click the name of the field that you want to edit.

   This opens the **Update extended field** 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).

2. 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.

3. 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 KUMA web console, in the **Settings** → **Extended event schema fields** section, click the **Import** button.
2. 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 KUMA web console, go to the **Settings** → **Extended event schema fields** section.
2. 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.
3. 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 KUMA web console, go to the **Settings** → **Extended event schema fields** section.
2. 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.
3. 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. Enrichment rules are stored in the settings of the normalizer where they were created.

You can create enrichment rules by clicking the **Add enrichment** button. To delete an enrichment rule, click ✕ next to it. Extended event schema fields can be used for event enrichment. Available enrichment rule settings are listed in the table below.

Available enrichment rule settings

Setting	Description

Source kind

Enrichment type. Depending on the selected enrichment type, advanced settings that will also need to be completed will be displayed. 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 enrichment fields** setting 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 extended schema array 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. 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.
Mapping	Event fields for data transfer: • **Dictionary field** specifies dictionary fields from which data is to be transmitted. The available fields depend on the selected dictionary resource. • **KUMA field** specifies event fields to which data is to be transmitted. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written there.

  The first field in the table (**Dictionary field**) is taken as the key with which the fields selected from the event as key fields are matched (**KUMA field**). As the key in the **Dictionary field**, you must select an indicator of compromise by which the enrichment is to be performed, for example, IP address, URL, or hash. In the rule, you must select the event field that corresponds to the selected indicator of compromise in the dictionary field.

  If you want to select multiple key fields, you can specify them using `|` as a separator (when specifying in the web interface or importing as a CSV file), for example, `< IP address >|< user name >`.

  You can add new table rows or delete table rows. To add a new table row, click **Add new element**. To delete a row in the table, click 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 for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, 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.
- Converting the IP address to the IPv4 format:
  - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.
  - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

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**.

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.

  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 the result of 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}}
  ```

  Required setting.

Target field	The KUMA event field that you want to populate with the data. Required setting. This setting is not available for the enrichment source of the **Table** type.

Conditions for sending 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 **Use 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 **Use raw event** drop-down list.

  > The **Use raw event** option is not applied if the sql parsing method is used in the main event parsing rule.

- **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.

If, when configuring the additional normalizer, you set the **Use raw event** setting to **No**, and left the **Source field** blank, any conversions configured for the field that is passed to the normalizer will not work. For conversions to be performed correctly, the source field must be filled, or the **Use raw event** setting must be set to Use raw event.

KUMA does not limit or check the text that you specify in this setting.

- Set of filters—used to define complex conditions that must be met by the events received by the normalizer.

  You can use the **Add condition** button to add a string containing fields for identifying the condition (see below).

  You can use 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. KUMA does not limit or check the text that you specify in the left and right operand fields.

- 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. This opens the **Conversion** window, in which 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 for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, 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.

- Converting the IP address to the IPv4 format:

  - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.

  - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

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**.

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 are system logs, operation logs, connections, the IPS module.
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 events received from Alcatel network switches via Syslog.
Ansible	[OOTB] Ansible syslog	Syslog	Designed for processing events from Ansible 2.10.8 via Syslog.
Apache Cassandra	[OOTB] Apache Cassandra file	regexp	Designed for processing events from the logs of the Apache Cassandra database version 4.0.
Apache Kafka	[OOTB] Apache Kafka syslog	regexp	Designed for processing Apache Kafka 3.8.1 system audit events received via Syslog.
Apache Tomcat	[OOTB] Apache Tomcat json	Syslog	Designed for processing events from the Apache Tomcat system (version 9.0 and later, 10.1 and later, 11.0 and later) via Syslog in JSON and key-value formats.
APC Smart-UPS	[OOTB] APC Smart UPS	regexp	Designed for processing APC Smart-UPS system audit events.

System name	Normalizer name	Type	Normalizer description
Aruba ClearPass	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Atlassian Confluence	[OOTB] Atlassian Jira Conflunce 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 Conflunce 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 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 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.

System name	Normalizer name	Type	Normalizer description
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.
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.
Check Point	[OOTB] Checkpoint syslog	Syslog	Designed for processing events received from the Check Point 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 ACE	[OOTB] Cisco ASA and IOS syslog	regexp	Designed for processing audit events from Cisco ASA, FTD, FWSM, NGIPS, PIX, ACE network devices and devices running Cisco IOS via Syslog.
Cisco ASA	[OOTB] Cisco ASA and IOS syslog	Syslog	Designed for processing audit events from Cisco ASA, FTD, FWSM, NGIPS, PIX, ACE network devices and devices running Cisco IOS 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) 14.2 proxy server, the access.log file.
Cisco ESA syslog	[OOTB] Cisco ESA syslog	Syslog	Designed for processing certain types of events received from Cisco ESA 16.0 via Syslog.
Cisco Firepower Threat Defense	[OOTB] Cisco ASA and IOS syslog	Syslog	Designed for processing audit events from Cisco ASA, FTD, FWSM, NGIPS, PIX, ACE network devices and devices running Cisco IOS via Syslog.
Cisco Firewall Services Module (FWSM)	[OOTB] Cisco ASA and IOS syslog	Syslog	Designed for processing audit events from Cisco ASA, FTD, FWSM, NGIPS, PIX, ACE network devices and devices running Cisco IOS via Syslog.
Cisco Identity Services Engine (ISE)	[OOTB] Cisco ISE syslog	regexp	Designed for processing events of the Cisco Identity Services Engine (ISE) system v3.2 received via Syslog.
Cisco IOS	[OOTB] Cisco ASA and IOS syslog	Syslog	Designed for processing audit events from Cisco ASA, FTD, FWSM, NGIPS, PIX, ACE network devices and devices running Cisco IOS 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.

System name	Normalizer name	Type	Normalizer description
Cisco Next-Generation Intrusion Prevention System (NGIPS)	[OOTB] Cisco ASA and IOS syslog	Syslog	Designed for processing audit events from Cisco ASA, FTD, FWSM, NGIPS, PIX, ACE network devices and devices running Cisco IOS via Syslog.
Cisco PIX	[OOTB] Cisco ASA and IOS syslog	regexp	Designed for processing types of audit events from Cisco ASA, FTD, FWSM, NGIPS, PIX, ACE network devices and devices running Cisco IOS via Syslog.
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 Wireless LAN Controllers (WLCs)	[OOTB] Cisco WLC syslog	regexp	Designed for processing 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.
Cisco Web Security Appliance (WSA)	[OOTB] Cisco WSA file, [OOTB] Cisco WSA syslog	regexp	[OOTB] Cisco WSA file is a normalizer 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 is a normalizer 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	Contains a resource set for processing some of the events of the SIP module of the CommuniGate Pro system versions 6.1 and 6.5.1, transmitted by the KUMA agent via TCP. For details on configuring the collector, see Creating a collector.
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	[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.

System name	Normalizer name	Type	Normalizer description
Cyberprotect Cyber Backup	[OOTB] Cyberprotect Cyber Backup SQL [OOTB] Cyberprotect Cyber Backup syslog	sql, regexp	[OOTB] Cyberprotect Cyber Backup SQL is a normalizer designed to process events received by the connector from the database of the Cyber Backup system (version 16.5). [OOTB] Cyberprotect Cyber Backup syslog is a normalizer designed to process events received from the Cyber Backup system (version 17.2, 17.3) via Syslog in CEF format.
Deep Instinct	[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 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.
D-Link Switches	[OOTB] D-Link Switches	regexp	Designed for processing events from D-Link switches (DES-2108, DES-3200-28, DES-3828, DES-3800) via Syslog.
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.
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 Threat Intelligence Platform	[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] ESET Protect syslog CEF	Syslog	Designed for processing events received in the CEF format received from the ESET PROTECT system (version 11.0) via Syslog. This package is available for KUMA version 3.2 or later.
Extreme Networks Summit Wireless Controller	[OOTB] Extreme Networks Summit Wireless Controller	regexp	Designed for processing of 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 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.

System name	Normalizer name	Type	Normalizer description
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.
Flant Deckhouse	[OOTB] Flant Deckhouse json	json	Designed for processing Metadata level audit events from Deckhouse (version 1.67) in JSON 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] FortiGate syslog KV [OOTB] FortiGate syslog CEF	regexp	The [OOTB] FortiGate syslog KV normalizer is designed for processing events received from FortiGate firewalls (version 7.0, 7.2) via syslog. The event source is FortiGate logs in key-value format. The [OOTB] FortiGate syslog CEF normalizer is designed for processing events received from Fortinet FortiGate (FortiOS 6.4) via syslog in CEF 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 FortiPAM	[OOTB] Fortinet FortiPAM syslog	regexp	Designed for processing events from Fortinet FortiPAM system (version 7.4) in key-value format via Syslog.
Fortinet FortiSOAR	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Fortinet FortiWeb	[OOTB] Syslog-CEF	Syslog	Designed for processing events from the Fortinet FortiWeb 7.6.1 system in 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
FreeIPA	[OOTB] FreeIPA	json	Designed for processing events from the FreeIPA system. The event source is FreeIPA 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 Garda DB	[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.

System name	Normalizer name	Type	Normalizer description
Grafana	[OOTB] Grafana syslog json	Syslog	Designed for processing events from Grafana 9.5.15 in JSON format via Syslog.
HAProxy	[OOTB] HAProxy syslog	Syslog	Designed for processing events received from HAProxy 2.4 via Syslog. This normalizer supports events in the Default, TCP, TCP CLF (version 3.2 and later), HTTP, HTTP CLF, HTTPS (2.6 and later), and Error formats.
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	Designed for processing of 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 Huawei VRP 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 Privileged Access Manager	[OOTB] Indeed PAM syslog [OOTB] Indeed PAM syslog CEF	Syslog	The [OOTB] Indeed PAM syslog normalizer is designed to process events received from the Indeed Privileged Access Manager version 2.6. The [OOTB] Indeed PAM syslog CEF normalizer is designed to process events received from the Indeed Privileged Access Manager (version 3.3) in the CEF format via Syslog.
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.
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	Normalizer for Juniper JUNOS (version 24.2) events received via Syslog.
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.

System name	Normalizer name	Type	Normalizer description
Kaspersky DFI	[OOTB] Kaspersky DFI json	json	Designed for processing audit events of the Kaspersky DFI system in JSON format. The normalizer supports the processing of events generated by the "dfi" KUMA connector. Available in the resource set for KUMA 4.0 and later.
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.
Kaspersky Industrial CyberSecurity for Networks 4.2	[OOTB] Kaspersky Industrial CyberSecurity for Networks 4.2 syslog [OOTB] Kaspersky Industrial CyberSecurity for Networks 4.x syslog	Syslog	The [OOTB] Kaspersky Industrial CyberSecurity for Networks 4.2 syslog normalizer is designed to process events received from the Kaspersky Industrial CyberSecurity for Networks 4.2 system via Syslog. This normalizer is deprecated. We recommend using the [OOTB] Kaspersky Industrial CyberSecurity for Networks 4.x syslog normalizer.
Kaspersky Industrial CyberSecurity for Networks 4.2, 4.3, 4.5	[OOTB] Kaspersky Industrial CyberSecurity for Networks 4.x syslog	Syslog	Designed for processing events received from Kaspersky Industrial CyberSecurity for Networks (versions 4.2, 4.3, 4.5) via Syslog.
Kaspersky IoT Secure Gateway (KISG)	[OOTB] Kaspersky KISG syslog	Syslog	Designed for processing events received from Kaspersky IoT Secure Gateway (KISG) 3.0 via Syslog.
Kaspersky NGFW	[OOTB] Kaspersky NGFW CEF	cef	Designed for processing events received from the Kaspersky NGFW system version 1.0, 1.2 in CEF format.
Kaspersky NDR	[OOTB] Kaspersky NDR syslog	Syslog	Designed for processing events received from the Kaspersky NDR 7.0 system via Syslog. This package is available for KUMA version 3.2 or later.
Kaspersky Security Center	[OOTB] KSC	cef	Designed for processing Kaspersky Security Center events received in CEF format.
Kaspersky Security Center	[OOTB] KSC from SQL	sql	Designed for processing events received by the connector from the database of the Kaspersky Security Center 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	Designed for processing Kaspersky Security for Exchange (KSE) 9.0 events stored in the database.
Kaspersky Secure Mail Gateway (KSMG)	[OOTB] KSMG syslog CEF [OOTB] KSMG 2.1+ syslog CEF	Syslog	[OOTB] KSMG syslog CEF is a normalizer for processing KSMG 2.0 events received in CEF format via Syslog. [OOTB] KSMG 2.1+ syslog CEF is a normalizer for processing KSMG 2.1.1, 3.0 events received in CEF format via Syslog.
Kaspersky Unified Monitoring and Analysis Platform (KUMA)	[OOTB] KUMA forwarding	json	Designed to process events forwarded from Kaspersky Unified Monitoring and Analysis Platform (KUMA).

System name	Normalizer name	Type	Normalizer description
Kaspersky Web Traffic Security (KWTS)	[OOTB] KWTS Syslog CEF, [OOTB] KWTS (KV)	Syslog	The [OOTB] KWTS Syslog CEF normalizer is designed for processing events received from Kaspersky Web Traffic Security in CEF format via Syslog. The [OOTB] KWTS (KV) normalizer is designed for processing Kaspersky Web Traffic Security events in key-value format.
Keycloak	[OOTB] Keycloak syslog	Syslog	Designed for processing events of the Keycloak 26.3 system received via Syslog.
Kemp Technologies 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.
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. This package is available for KUMA version 3.2 or later.
McAfee Web Gateway	[OOTB] McAfee Web Gateway	regexp	Designed for processing events from McAfee Web Gateway 7.8 and later in CEF and KV formats received via Syslog.
Microsoft Active Directory Certificate Services (AD CS)	[OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	xml	Designed for processing Active Directory Certificate Services (AD CS) log events in XML format. The [OOTB] Microsoft Products for KUMA 3 normalizer supports this event source in KUMA 3.0.1 and later versions.

System name	Normalizer name	Type	Normalizer description
Microsoft Active Directory Federation Service (AD FS)	[OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	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 [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	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 [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	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 [OOTB] Microsoft DNS ETW logs json	regexp	The [OOTB] Windows DNS normalizer is designed to process 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. The [OOTB] Microsoft DNS ETW logs json normalizer is designed to process some Microsoft DNS Server audit events supplied by the ETW provider. This package is available for KUMA version 3.2 or later.
Microsoft Exchange Server	[OOTB] Exchange CSV	csv	Designed for processing the event log of the Microsoft Exchange system. The event source is Exchange server MTA logs.

System name	Normalizer name	Type	Normalizer description
Microsoft Hyper-V	[OOTB] Microsoft Products [OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	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 Installer	[OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	xml	Designed for processing Microsoft Installer (MSI Installer) log events in XML format.
Microsoft IIS	[OOTB] IIS Log File Format	regexp	Designed for processing events from in the format described at https://learn.microsoft.com/en-us/windows/win32/http/iis-logging. The event source is Microsoft IIS logs.
Microsoft Network Policy Server (NPS)	[OOTB] Microsoft Products [OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	xml	Designed for processing Microsoft Windows events. The event source is Network Policy Server events.
Microsoft Office 365	[OOTB] Microsoft Office 365 json	json	Designed for processing Microsoft Office 365 audit events. This normalizer supports processing events received from Microsoft Teams, Azure Active Directory, SharePoint systems. This package is available for KUMA version 3.4 or later.
Microsoft SCCM	[OOTB] Microsoft SCCM file	regexp	Designed for processing events of the Microsoft SCCM system version 2309. This normalizer supports processing of some of the events stored in the AdminService.log file.

System name	Normalizer name	Type	Normalizer description
Microsoft SharePoint Server	[OOTB] Microsoft SharePoint Server diagnostic log file	regexp	Designed for processing some 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 [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	xml	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 [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	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 → AppLocker	[OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	xml	Designed for processing Microsoft → Windows → AppLocker log events in XML format.

System name	Normalizer name	Type	Normalizer description
Microsoft → Windows → DNS-Server → Audit	[OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	xml	Designed for processing Microsoft → Windows → DNS-Server → Audit log events in XML format.
Microsoft → Windows → Group Policy → Operational	[OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	xml	Designed to process Microsoft → Windows → Group Policy → Operational log events in XML format.
Microsoft → Windows → PowerShell	[OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	xml	Designed for processing Microsoft Windows PowerShell log events.
Microsoft → Windows → PowerShell-Operational	[OOTB] Microsoft Products [OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	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.

System name	Normalizer name	Type	Normalizer description
Microsoft → Windows → System log → Netlogon	[OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	xml	Designed for processing Microsoft → Windows → System log → Netlogon log events in XML format.
Microsoft → Windows → TaskScheduler → Operational	[OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	xml	Designed for processing Microsoft → Windows → TaskScheduler → Operational log events in XML format.
Microsoft → Windows → WinRM → Operational	[OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	xml	Designed for processing Microsoft → Windows → WinRM → Operational log events in XML format.
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 SQL Server, Microsoft SQL Server Express	[OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	xml	Designed to process events of MS SQL Server 2008 or newer.

System name	Normalizer name	Type	Normalizer description
Microsoft Windows Remote Desktop Services	[OOTB] Microsoft Products [OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	xml	Designed for processing Microsoft Windows events. Event source → Applications and Services Logs → Microsoft → Windows → TerminalServices-LocalSessionManager → Operational log 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 Service Control Manager	[OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	xml	Designed for processing events from the Service Control Manager logs (System log) of the Microsoft Windows operating system.
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 [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	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.
Microsoft WSUS	[OOTB] Microsoft WSUS file	regexp	Designed for processing events of the Gardatech Perimeter system version 5.3, 5.4 received via Syslog.

System name	Normalizer name	Type	Normalizer description
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 some 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 is a normalizer designed for processing events of the NetApp system (ONTAP 9.12 version) received via Syslog. [OOTB] NetApp file is a normalizer designed for processing events of the NetApp system (ONTAP 9.12 version) 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). This 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. This normalizer supports processing some of the events from the C:\Program Files\NetApp\SnapCenter WebApp\App_Data\log\audit.log file.
Netgear Switch	[OOTB] Netgear Switch syslog	regexp	Designed for processing audit events from Netgear switches (model M4300-52G).
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.
Netwrix Endpoint Protector	[OOTB] Netwrix Endpoint Protector syslog	regexp	Designed for processing events received from the Netwrix Endpoint Protector 5.9 system via Syslog. This package is available for KUMA version 3.2 or later.
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.
NFS kernel server	[OOTB] NFS kernel server syslog	regexp	Designed for processing NFS kernel server events (NFSv3, NFSv4.0, NFSv4.1, NFSv4.2) in Syslog format.
Nginx	[OOTB] nginx regexp	regexp	Designed for processing the event log of the nginx web server in the Combined format. The normalizer supports the processing of events received via Syslog and stored in a file. This package is available for KUMA version 3.2 or later.
NIKSUN NetDetector	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
NPO Fractel	[OOTB] NPO Fractel SSPT-4	regexp	Designed for processing events of the NPO Fractel SSPT-4A1 system.
One Identity Privileged Session Management	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.

System name	Normalizer name	Type	Normalizer description
OpenLDAP	[OOTB] OpenLDAP	regexp	Designed for line-by-line processing of some events of the OpenLDAP 2.5 system in an auditlog.ldif file.
OpenVPN	[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.
Orion soft Cloudlink	[OOTB] OrionSoft Cloudlink syslog	regexp	Designed for processing events from the Orion soft Cloudlink 1.0 system in CEF format received via Syslog.
Orion soft Termit	[OOTB] OrionSoft Termit syslog	Syslog	Designed for processing events received from the Orion soft Termit 2.2 system via Syslog.
Orion soft zVirt	[OOTB] Orion Soft zVirt syslog	regexp	Designed for processing events of the Orion soft zVirt virtualization system (zVirt 3.1, 4.4, 4.5, 5.0 and zVirt Max 1.2).
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] Palo Alto 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 Multiscanner	[OOTB] PTsecurity Multiscanner json	Syslog	Designed for processing events from PT Multiscanner 5.24 in JSON format received via Syslog.
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	Syslog	Designed for processing events from PTsecurity Web Application Firewall 4 received via Syslog.
Postfix	[OOTB] Postfix syslog	regexp	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.

System name	Normalizer name	Type	Normalizer description
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.
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.
RuPost LLC WorksPad	[OOTB] RuPost LLC WorksPad CEF syslog	Syslog	Designed for processing part of the events of the RuPost LLC WorksPad system (version 6.3.70) in CEF format via Syslog.
SafeTech CA SQL	[OOTB] SafeTech CA SQL	sql	Designed to process events from the PostgreSQL DBMS of the SafeTech CA 3.2 system.
SailPoint IdentityIQ	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
SaltStack	[OOTB] SaltStack syslog	Syslog	Designed for processing single-line events from the SaltStack system (version 3004.1 and later) received via Syslog. The normalizer does not support multi-line events.
SecurityCode Continent 3.9	[OOTB] SecurityCode Continent 3.9 json	json	Designed for processing SecurityCode Continent 3.9.2 events received from the kuma-kont utility in json format. This package is available for KUMA version 3.4 or later.
SecurityCode Continent 4	[OOTB] SecurityCode Continent 4 syslog	regexp	Designed for processing events of the SecurityCode Continent system version 4 received via Syslog.

System name	Normalizer name	Type	Normalizer description
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.
Seqrite EPS	[OOTB] Seqrite EPS CEF	cef	Designed for processing events from the Seqrite EPS 6.0 system in CEF format.
Skype for Business	[OOTB] Microsoft Products for KUMA 3 [OOTB] Microsoft Products via KES WIN NG for KUMA 3.2 [OOTB] Microsoft Products via KES WIN NG for KUMA 4.0.1	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	Syslog	Designed to process events from the Sonicwall TZ firewall in key-value format (version 7.3), and in CEF format (versions 6.5, 7.3) received via Syslog.
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	Designed for 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.
SolarWinds SFTP & SCP Server file	[OOTB] SolarWinds SFTP & SCP Server file	regexp	Designed for processing the SolarWinds SFTP & SCP Server 2.2 event log stored in the SftpServer.log file. This package is available for KUMA version 3.2 or later.
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
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	Designed for processing Suricata 7.0.1 events stored in a JSON file. This 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.
Synesis The Wall WAF	[OOTB] Synesis The Wall WAF syslog	Syslog	Designed for processing events of the Synesis TheWall WAF system received via Syslog.
TeamPass	[OOTB] TeamPass SQL	sql	Designed for processing events from the TeamPass 3.1 system stored in the database.

System name	Normalizer name	Type	Normalizer description
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	Designed for processing some of the Tionix VDI system (version 2.8) events stored in the tionix_Intmov.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.
Umbraco CMS	[OOTB] Umbraco CMS syslog	regexp	Designed for processing events of the Umbraco CMS version 11.4.1 in Syslog 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.
Veeam Software – Veeam Backup & Replication	[OOTB] Veeam Software Veeam Backup & Replication syslog	regexp	Designed for processing events from the Veeam Software – Veeam Backup & Replication 12.1 system via Syslog.
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.
VK Private Cloud	[OOTB] VK Private Cloud syslog	regexp	Designed for processing audit events of the VK Private Cloud 1.0 system received via Syslog.
VK WorkSpace Mail	[OOTB] VK WorkSpace Mail syslog	Syslog	Designed for processing events received from the VK WorkSpace Mail 1.23 system via Syslog in key-value format.
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.

System name	Normalizer name	Type	Normalizer description
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.
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	Syslog	Designed for processing WatchGuard Firebox (version OS v2026.1.B730198) events received via Syslog.
Webmonitorx	[OOTB] Webmonitorx json	regexp	Designed for processing events coming from the Webmonitorx system in JSON format.
Webroot BrightCloud	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Windchill FRACAS	[OOTB] PTC Winchill Fracas	regexp	Designed for processing events of the Windchill FRACAS failure registration system.
Yandex Cloud	[OOTB] Yandex Cloud	regexp	Designed for processing a part of Yandex Cloud audit events. This 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), Backup.
Zabbix	[OOTB] Zabbix SQL	sql	Designed for processing events of Zabbix 6.4.
Zecurion DLP	[OOTB] Zecurion DLP syslog	regexp	Designed to process a part of events of the Zecurion DLP system version 13.0 received in CEF format 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 – Synonyx	[OOTB] Bastion Synonyx CEF	regexp	Designed for processing events in the CEF format received from the IT-Bastion Synonyx 1.7 system. This normalizer package is available for KUMA version 3.2 or later.
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.
Angie web server	[OOTB] Angie syslog	Syslog	Designed for processing events received from Angie and Angie Pro (version 1.10.1) via Syslog. This normalizer package is available for KUMA version 3.2 or 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`

System name	Normalizer name	Type	Normalizer description
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 [OOTB] ViPNet Coordinator CEF	Syslog	The [OOTB] ViPNet Coordinator Syslog normalizer is designed to process events from the InfoTeCS ViPNet Coordinator system via Syslog. The [OOTB] ViPNet Coordinator CEF normalizer is designed to process events received from the InfoTeCS ViPNet Coordinator (version 4.5) in CEF format.
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 to process events from systems: Dallas Lock 8 (version 11.4.2.843, 12.10.6.877), Dallas Lock Unified Control Center (ECU) (version 4.0.294.0, 4.0.354.0) in CEF format.
Confident - Dallas Lock	[OOTB] Confident Dallas Lock syslog CEF	regexp	Designed for processing events received from Dallas Lock 8 (version 11.4.2) 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	Designed for processing 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	Syslog	Normalizer for processing system events of the SKDPU NT Access gateway 7.0 (build 17, update 14) and 7.1.1 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.
Yandex 360	[OOTB] Yandex 360 json	json	Designed for processing events received from the Yandex 360 system in JSON format. The transport for receiving events will be available in KUMA 4.2.
Yandex Browser corporate	[OOTB] Yandex Browser	json	Designed for processing events received from the corporate version of Yandex Browser 23, 24.4, 25.2.
-	[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 the 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 **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 collector settings. Available aggregation rule settings are listed in the table below.

General tab

General tab

Setting	Description
Name	Unique name of the resource. Maximum length of the name: 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
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`.
Triggered rule lifetime	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`. Required setting.
Description	Description of the resource. Maximum length of the description: 4000 Unicode characters.
Identical fields	Fields of normalized events whose values must match. For example, for network events, `SourceAddress`, `DestinationAddress`, and `DestinationPort` normalized event fields can be used. In the aggregation event, these normalized event fields are populated with the values of the base events. Required setting.
Unique fields	Normalized event fields whose values must be unique for those events to be considered in an aggregated event. If the value of a field specified under **Unique Fields** is duplicated in multiple events, only one of such events is included in the aggregated event, and the duplicates are ignored. If you specify more than one field under **Unique Fields**, the combinations of values of all of these fields must be unique. Unique fields are not saved in the aggregated event.
Sum fields	Fields whose values are summed up during aggregation and written to the same-name fields of the aggregated event. The following special considerations are relevant to field behavior: • The values of fields of the `Number` and `Float` types are summed up. • The values of fields of the `String` type are concatenated with commas added as separators. • The values of fields with the types `Array of strings`, `Array of numbers` and `Array of floats` are appended to the end of the array.

Setting	Description
Filter	Conditions for determining which events must be processed by the resource. In the drop-down list, you can select an existing filter **Create new** to 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** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

Setting	Description
	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 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.

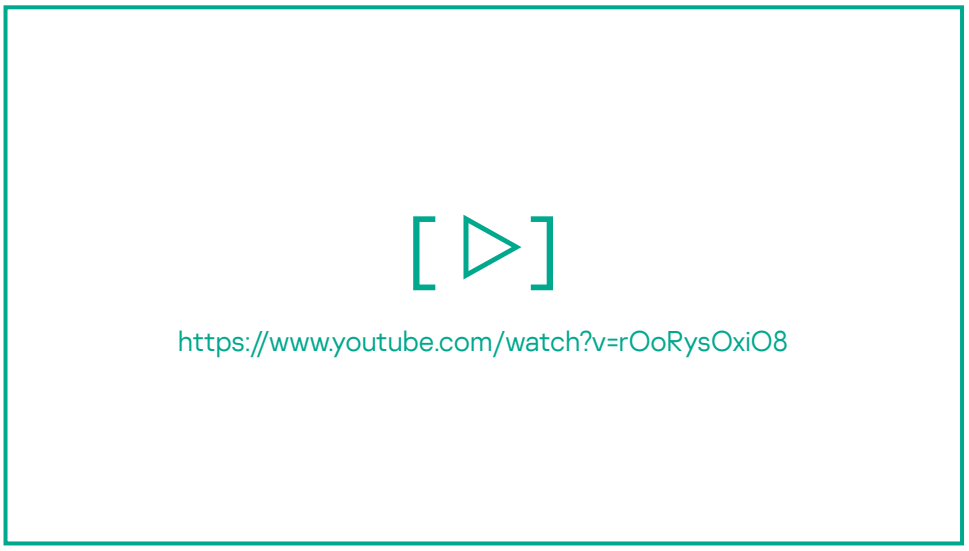
Predefined aggregation rules

The **[OOTB] Netflow 9** aggregation rule is included in the KUMA distribution kit. The rule is triggered after 100 events or 10 seconds.

Events are aggregated by the following 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. In the collector, you can create an enrichment rule, and it becomes a resource that you can reuse in other services. You can also link an enrichment rule created as a standalone resource.
- Correlator. In the correlator, you can create an enrichment rule, and it becomes a resource that you can reuse in other services. You can also link an enrichment rule created as a standalone resource.
- Normalizer. In the normalizer, you can only create an enrichment rule linked to that normalizer. Such a rule will not be available as a standalone resource for reuse in other services.

Available enrichment rule settings are listed in the table below.

General tab

This tab lets you specify the general settings of the enrichment rule (see the table below).

Setting	Description

Source kind

Required setting.

Drop-down list for selecting the type of incoming events. Depending on the selected type, the following additional settings will be displayed:

- **template**

  This type of enrichment is used when you need to write the result of 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}}
  ```

- **table**

  This type of enrichment is used if you need to add a value from the dictionary of the **Table** 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.
Mapping	Event fields for data transfer: • **Dictionary field** specifies dictionary fields from which data is to be transmitted. The available fields depend on the selected dictionary resource. • **KUMA field** specifies event fields to which data is to be transmitted. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written there.

  The first field in the table (**Dictionary field**) is taken as the key with which the fields selected from the event as key fields are matched (**KUMA field**). As the key in the **Dictionary field**, you must select an indicator of compromise by which the enrichment is to be performed, for example, IP address, URL, or hash. In the rule, you must select the event field that corresponds to the selected indicator of compromise in the dictionary field.

  If you want to select multiple key fields, you can specify them using `|` as a separator (when specifying in the web interface or importing as a CSV file), for example, `< IP address >|< user name >`.

  You can add new table rows or delete table rows. To add a new table row, click **Add new element**. To delete a row in the table, click the ✕ button.

- **enrichment by assets**

  You can use this type of enrichment when you need to add enrichment of events with assets to specific collectors.

  To enrich events with assets under **Mapping**, fill in the mapping table of the KUMA event fields and the fields used to record values from the hash table:

  1. Click **Add**.
  2. In the **KUMA field** drop-down list, select one or more event fields that you want to map to values from the hash table and write to the corresponding asset information field in the table.
  3. In the **KUMA event field to write to** drop-down list, select the field into which you want to write the event ID, if the event field (**KUMA field**) matches the value from the hash table.

     Possible parameter values in the field mapping table
  ```

KUMA field	KUMA event field to write to
DestinationAddress DestinationHostName DeviceAddress DeviceHostName SourceAddress SourceHostName DestinationMacAddress DeviceMacAddress SourceMacAddress DeviceCustomString1 DeviceCustomString2 DeviceCustomString3 DeviceCustomString4 DeviceCustomString5 DeviceCustomString6	SourceAssetID DeviceAssetID DestinationAssetID

If the user selects multiple fields, the **AND** operator is applied to field values. If the user adds multiple strings in the **KUMA field** column, the **OR** operator is applied to field values. The matching begins from the first field in the list. If the value of the field matches the value of the attribute, matching stops.

For example, suppose that the following data is received in an event:

```
SourceAddress=192.168.x.xx
SourceHostName=kuma.example.com.
```

There are two assets in KUMA: *company.example.com* (IP address: 192.168.x.xx, FQDN: company.example.com) and *kuma.example.com* (IP address: 192.168.y.yyy, FQDN: kuma.example.com). Matching is performed as follows:

- If the user separately specified `sourceAddress` → `sourceAssetID`, and then `sourceHostname` → `sourceAssetID`, then the asset matches the first condition (company.example.com).
- If it is more important for the user to get the `sourceHostname` value (kuma.example.com), then `sourceHostname` → `sourceAssetID` must be specified first, followed by `sourceAddress` → `sourceAssetID`.
- If the user specifies `sourceAddress` + `sourceHostname` → `sourceAssetID`, enrichment does not occur.

At least one matching condition must be specified in the field mapping table.

You can also click the **Apply default mapping** button above the table. In this case, the values of **KUMA field** and **KUMA event field to write to** are applied.

Default values of parameters in the field mapping table

KUMA field	KUMA event field to write to
DestinationAddress	DestinationAssetID
DestinationHostName	DestinationAssetID
DeviceAddress	DeviceAssetID
DeviceHostName	DeviceAssetID
SourceAddress	SourceAssetID
SourceHostName	SourceAssetID

- **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 enrichment fields** setting 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 extended schema array 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.
  - Under **Conversion**, 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 for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, 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.
    - Converting the IP address to the IPv4 format:
      - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.
      - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

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**.

- **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** is the number of workers that the service can run simultaneously to process events in parallel. Specify a positive integer number from 0 to 999. If 0 is specified, the number of logical CPU cores is used.
  - **Max tasks**—maximum number of simultaneously fulfilled requests. By default, this value is equal to the number of vCPUs of the KUMA Core server.
  - **Cache TTL**—the lifetime (in seconds) of the values stored in the cache. The default value is `60` (seconds).
  - **Cache disabled**—you can use this toggle switch to enable or disable caching. Caching is enabled by default.
  - **Cache update disabled**—if cache update is enabled, then after the cache TTL expires, the collector sends repeat queries to the DNS server to update the information stored in the cache. This setting is available only when cache is enabled. By default, cache update is enabled.
  - The **Recursive queries** setting is available starting with KUMA 3.4.1. You can use this toggle switch to make a KUMA collector send recursive queries to authoritative DNS servers for the purposes of enrichment. The default value is **Disabled**.

- **cybertrace**

  For systems with a high load on the collector or correlator, we recommend using the cybertrace-http enrichment type. This type returns a structured response in JSON format and correctly distinguishes between strings and arrays of custom attributes, preventing parsing errors.

  When enriched through cybertrace, the collector gets an unstructured response. If the value of a custom attribute in CyberTrace contains square brackets, for example, **[str1**, **str2]**, KUMA treats it as an array, not a string.

  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.
  - **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 is a new streaming event enrichment type in CyberTrace that allows you to send a large number of events with a single request to the CyberTrace API. Recommended for systems with a lot of events. The performance of cybertrace-http is superior to that of the cybertrace type.

Limitations:

- The cybertrace-http enrichment type cannot be used for retroscan in KUMA.
- If the cybertrace-http enrichment type is being used, detections are not saved in CyberTrace history in the Detections window.

Available settings:

- **Name** (required) is a unique name for the enrichment rule. Up to 128 Unicode characters.
- **Tenant** (required) is the name of the tenant in which you plan to work with CyberTrace.
- **URL** (required)—in this field, you can specify the URL of a CyberTrace server to which you want to send requests and the port that CyberTrace API is using. The default port is 443.
- **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 1,000,000. 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.
- **Major version of CyberTrace** selects 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**.
- The **Debug** toggle switch enables resource logging. The toggle switch is turned off by default.
- **Tags** for resource search.

- **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, under **Mapping geographic data to event fields**, 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 pre-configured 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.

- **check DLL Hijacking**

  DLL Hijacking is an attack technique that involves delivering vulnerable legitimate software along with a malicious dynamic link library (DLL) to the target system. 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 status, KUMA sends a request to KSN with information from event fields. In response to the request, KSN sends the analysis result.

  You can configure DLL Hijacking attack detection in the collector or correlator.

  > To use enrichment such as **check DLL Hijacking**, your license must include the AI module. The General administrator must also accept the additional KSN Statement in the **Kaspersky Security Network** section (**Settings** → **Integrations** → **AI services** → **DLL Hijacking detection**).

  Available settings:

  - **Workers** is the number of workers that the service can run simultaneously to process events in parallel. Specify a positive integer number from 0 to 999. If 0 is specified, the number of logical CPU cores is used.
  - **Maximum cache size** is the maximum cache size for KSN responses, in megabytes. The default value is `128 MB`.

    Each KSN response has its own lifetime, during which it is retained in the cache. If KUMA sends a second request for a DLL file for which KSN has already issued a response, and the lifetime of that response has not yet expired, the request is not actually sent to KSN, but the response is taken from the cache.
  - If the **Maximum number of events in enrichment queue** is exceeded, new events are not processed. The default value is `1,000,000`. `0` means that event processing stops if the queue contains any number of events.
  - **Perform a repeat request**. KSN may return the **Unclassified** status. This status means that KSN has accepted the request and is currently processing it, but there is no status yet. Processing may take a long time, so the request must be re-sent. If this setting is enabled, specify additional settings:
    - If the **Create duplicates** setting is enabled, KUMA creates a duplicate event and uses it for repeated requests to KSN. In this case, the main event is routed in accordance with the workflow. A duplicate event is written to the database when one of the following conditions is satisfied:
      - A status other than **Unclassified** is received from KSN.
      - The disk buffer configured using the **Buffer size** setting is full.
      - The **Request count** is exceeded.

	If this setting is disabled, KUMA continues to send requests using the main event. The event is written to the database when one of the following conditions is satisfied: • A status other than **Unclassified** is received from KSN. • The disk buffer configured using the **Buffer size** setting is full. • The **Request count** is exceeded. • The **Request count** is the maximum number of requests to KSN. If, after the specified number of requests, a status other than **Unclassified** is not received, the event is written to the database with the **Unclassified** status. The default value is `10`. • **Poll interval** is the interval between requests, in minutes. The default value is `1 minute`. The minimum value is `1 minute`. • **Buffer size** is the size of the disk buffer for events awaiting a repeat request, in megabytes. The default value is `10240 MB`. The minimum value is `10 MB`. • **Proxy** specifies proxy server settings, if one is needed to connect to the KSN server. You can select one of the available proxy servers or create a new proxy server. By default, a proxy server is not used. • **Mapping** is a table of parameters that must be sent in a KSN request to get a status, and a list of the corresponding KUMA event fields. List of parameters in the request: • **dll hash (md5)**. Recommended field for mapping: OldFileHash. • **dll path**. Full path including the file name; recommended field for mapping: OldFilePath. The path may contain personal data, so KUMA sends this parameter to KSN in an anonymized form. For example: Path before anonymization: `C:\Windows\Device\HarddiskVolume1\Users\Alex\malware.dll` The same path in anonymized form: `device\harddiskvolume1\users\username` • **process hash (md5)**. Recommended field for mapping: FileHash. • **process path**. Full path including the file name. The path may contain personal data, so KUMA sends this parameter to KSN in an anonymized form. For example: Path before anonymization: `C:\Windows\Device\HarddiskVolume1\Users\Alex\malware.dll` The same path in anonymized form: `device\harddiskvolume1\users\username` • **dll name**. Recommended field for mapping: OldFileName. • **process name**. Recommended field for mapping: FileName. • **process signature**. Optional; recommended field for mapping: FileID. In this field, you need to specify the KUMA field that passes the Organization from the Subject of the signature. If these values are not available, leave the signature empty. To add this parameter, click the **Add** button, then in the **DLL Hijacking field** field, select **process signature** from the drop-down list. • **dll size**. Optional; recommended field for mapping: OldFileSize. To add this parameter, click the **Add** button, then in the **DLL Hijacking field** field, select **dll size** from the drop-down list. For each parameter, specify a corresponding KUMA event field in the table.
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

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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

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 KUMA distribution kit includes the "[OOTB] KATA alert" enrichment rule. This rule is used to enrich events received from KATA in the form of a hyperlink to an alert. The hyperlink is put in the **DeviceExternalId** field.

If you want the **DeviceExternalID** value to be displayed as a link in the event properties, make sure that the **Template** field for the `DeviceAddress` parameter contains a URL or fully qualified domain name followed by the "/" character, for example: `https://example.com:8443/` or `https://example.com/`.

Configuring the enrichment of collectors with assets

If asset enrichment is performed on each collector, asset data (UUID, IP address, FQDN) has to be sent to each collector. This can significantly complicate and slow down the processing of data by collectors.

You can configure the enrichment of collectors with asset data only on the collectors that you want to be enriched.

To create an asset enrichment rule for a collector:

1. Go to the **Resources** section of the KUMA web interface.

   A table with the created enrichment rules is displayed in the right part of the **Resources** section.

2. In the list of tenants in the left part of the window, select the tenant that owns the resource.

3. Click the **Create new** button above the table.

4. In the displayed sidebar, specify the following settings:

   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 enrichment rule.

c. In the **Source kind** drop-down list, select **enrichment by assets**.

d. If necessary, use the **Debug** toggle switch to enable logging of service operations.

e. If necessary, in the **Tags** drop-down list, select thetags for the enrichment rule.

f. If required, add a description for the enrichment rule in the **Description** field. The rule description can contain up to 4000 Unicode characters.

5. In the **Mapping** section, populate the field mapping table as follows:

a. Click **Add**.

b. In the **KUMA field** drop-down list, select one or more event fields that you want to map to values from the hash table and write to the corresponding asset information field in the table.

c. In the **KUMA event field to write to** drop-down list, select the field into which you want to write the event ID, if the event field (**KUMA field**) matches the value from the hash table.

Possible parameter values in the field mapping table

KUMA field	KUMA event field to write to

`DestinationAddress`
`DestinationHostName`
`DeviceAddress`
`DeviceHostName`
`SourceAddress`
`SourceHostName`
`DestinationMacAddress`
`DeviceMacAddress`
`SourceMacAddress`
`DeviceCustomString1`
`DeviceCustomString2`
`DeviceCustomString3`
`DeviceCustomString4`
`DeviceCustomString5`
`DeviceCustomString6`
	`SourceAssetID`
	`DeviceAssetID`
	`DestinationAssetID`

If the user selects multiple fields, the **AND** operator is applied to field values. If the user adds multiple strings in the **KUMA field** column, the **OR** operator is applied to field values. The matching begins from the first field in the list. If the value of the field matches the value of the attribute, matching stops.

For example, suppose that the following data is received in an event:

```
SourceAddress=192.168.x.xx
SourceHostName=kuma.example.com.
```

There are two assets in KUMA: *company.example.com* (IP address: 192.168.x.xx, FQDN: company.example.com) and *kuma.example.com* (IP address: 192.168.y.yyy, FQDN: kuma.example.com). Matching is performed as follows:

- If the user separately specified `sourceAddress` → `sourceAssetID`, and then `sourceHostname` → `sourceAssetID`, then the asset matches the first condition (company.example.com).
- If it is more important for the user to get the `sourceHostname` value (kuma.example.com), then `sourceHostname` → `sourceAssetID` must be specified first, followed by `sourceAddress` → `sourceAssetID`.
- If the user specifies `sourceAddress` + `sourceHostname` → `sourceAssetID`, enrichment does not occur.

At least one matching condition must be specified in the field mapping table.

You can also click the **Apply default mapping** button above the table. In this case, the values of **KUMA field** and **KUMA event field to write to** are applied.

Default values of parameters in the field mapping table

KUMA field	KUMA event field to write to
DestinationAddress	DestinationAssetID
DestinationHostName	DestinationAssetID
DeviceAddress	DeviceAssetID
DeviceHostName	DeviceAssetID
SourceAddress	SourceAssetID
SourceHostName	SourceAssetID

6. Under **Filter parameters**, 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.

   Adding conditions and condition groups for the filter

   To adding conditions and condition groups for the filter:

   1. When creating an enrichment rule, make sure that under **Filter parameters**, **Builder** is selected.
   2. Click the **Add condition** button and in the drop-down lists that appear one by one, select event values and the filter operator.

      You can add multiple conditions to a filter.
   3. If necessary, click the **Add group** button to add a group of conditions for the filter.

      You can add multiple condition groups.
   4. Click a logical operator (**AND**, **OR**, **NOT**). **AND** is selected by default.

7. Click **Create**.

Applying an enrichment rule when managing collector settings

You can add one or more asset enrichment rules to an existing collector by managing its settings.

To add an asset enrichment rule in the collector settings:

1. In the KUMA web interface, under **Resources** → **Collectors**, select the relevant tenant.
2. Click a collector in the table of collectors.
3. In the left part of the displayed sidebar, select the **Event enrichment** section.
4. Click the **Add enrichment** button.
5. In the **Enrichment rule** drop-down list, select an asset enrichment rule for the collector.

   You can also create an asset enrichment rule by clicking **Create new** in the **Enrichment rule** drop-down list.

6. If necessary, add another asset enrichment rule by clicking the **Add enrichment** button.

   You cannot add the same enrichment rule more than once.

7. Click **Save**.

   KUMA applies the added enrichment rule when the collector receives and processes messages from event sources.

Data collection and analysis rules

Data mining rules are used for detecting incidents by analyzing events saved in the KUMA database.

In contrast to streaming correlation and retrospective scanning, data mining rules involve the preprocessing of data using SQL queries (including additional calculations, grouping and aggregation using SQL) and running regularly on a schedule. With correctly configured data mining rules, the SIEM can help you detect advanced threats.

To manage the section, you need one of the following roles: General administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.

When creating or editing data collection and analysis rules, you need to specify the settings listed in the table below.

Settings of data collection and analysis rules

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. If you have access to only one tenant, this field is filled in automatically. If you have access to multiple tenants, the name of the first tenant from your list of available tenants is inserted. You can select any tenant from this list.

SQL	Required setting.
	The SQL query must contain an aggregation function with a LIMIT and/or a data grouping with a LIMIT.
	You must use a LIMIT value between 1 and 10,000.
	Examples of SQL queries
	• A query containing only an aggregation function:
	`SELECT count(DeviceCustomFloatingPoint1) AS `Aggregate` FROM `events` WHERE Type = 1 ORDER BY Aggregate DESC LIMIT 10`
	• A query containing only data grouping:
	`SELECT SourceAddress, DeviceCustomFloatingPoint1 FROM `events` WHERE Type = 1 GROUP BY SourceAddress, DeviceCustomFloatingPoint1 ORDER BY DeviceCustomFloatingPoint1 DESC LIMIT 10`
	• A query containing an aggregation function and data grouping:
	`SELECT SourceAddress, sum(DeviceCustomFloatingPoint1) FROM `events` WHERE Type = 1 GROUP BY SourceAddress, DeviceCustomFloatingPoint1 ORDER BY DeviceCustomFloatingPoint1 DESC LIMIT 10`
	• A query containing an expression using aggregation functions:
	`SELECT stddevPop(DeviceCustomFloatingPoint1) + avg(DeviceCustomFloatingPoint1) AS `Aggregate` FROM `events` WHERE Type = 1 ORDER BY Aggregate DESC LIMIT 10`
	You can also use SQL function sets: `enrich` and `lookup`.

SQL query execution frequency	Required setting. The interval for executing the SQL query. You can specify the interval in minutes, hours, and days. The minimum interval is 1 minute. The default timeout of the SQL query is equal to the interval that you specify in this field. If the execution of the SQL query takes longer than the timeout, an error occurs. In this case, we recommend increasing the interval. For example, if the interval is 1 minute, and the query takes 80 seconds to execute, we recommend setting the interval to at least 90 seconds.
Tags	Optional setting. Tags for resource search.
Depth	Optional setting. Expression for the lower bound of the interval for searching events in the database. To select a value from the list or to specify the depth as a relative interval, place the cursor in the field. For example, if you want to find all events from one hour ago to now, set the relative interval of `now-1h`.
Description	Optional setting. Description of data collection and analysis rules.
Mapping	Settings of the mapping the fields of an SQL query result to KUMA events: **Source field** is the field from the SQL query result that you want to convert into a KUMA event. **Event field** is the KUMA event field. You can select one of the values in the list by placing the mouse cursor in this field. **Label** is a unique custom label for event fields that begin with DeviceCustom*. You can add new table rows or delete table rows. To add a new table row, click **Add mapping**. To delete a row, select the check box next to the row and click the button. If you do not want to fill in the fields manually, you can click the **Update mapping from SQL** button. The field mapping table is populated with the values of the SQL query fields, including aliases (if any). For example, if the value of an SQL query field is `SourceAddress` and this value is the same as the name of an event field, this value is inserted into in the **Event field** column in the field mapping table. Clicking the **Update mapping from SQL** button again does not refresh the table, and fields from the SQL query are added to it again.

You can create a data collection and analysis rule in one of the following ways:

- In the **Resources** → **Resources and services** → **Data collection and analysis rules** section.
- In the **Events** section.

*To create a data collection and analysis rule in the **Events** section:*

1. Create or generate an SQL query and click the button.

   A new browser tab for creating a data collection and analysis rule is opened in the browser with pre-filled **SQL query** and **Depth** fields. The field mapping table is also be populated automatically if you did not use an asterisk (*) in the SQL query.

2. Fill in the required fields.

   If necessary, you can change the value in the **SQL query execution frequency** field.

3. Save the settings.

   The data collection and analysis rule is saved and is available in the **Resources and services** → **Data collection and analysis rules** section. For a data collection and analysis rule to run, you must create a scheduler for it.

Configuring the scheduler for a data collection and analysis rule

For a data collection and analysis rule to run, you must create a scheduler for it.

The scheduler makes SQL queries to specified storage partitions with the interval and search depth configured in the rule, and then converts the SQL query results into base events, which it then sends to the correlator.

SQL query results converted to base events are not stored in the storage.

For the scheduler to work correctly, you must configure the link between the data collection and analysis rule, the storage, and the correlators in the **Resources** → **Data collection and analysis** section.

To manage this section, you need one of the following roles: General administrator, Tenant administrator, Tier 2 analyst, Read shared resources, Manage shared resources.

The schedulers are arranged in the table by the date of their last launch. You can sort the data in columns in ascending or descending order by clicking the down arrow (⌄) icon in the column heading.

Available columns of the table of schedulers:

- **Rule Name** is the name of the data collection and analysis rule for which you created the scheduler.
- **Tenant name** is the name of the tenant to which the data collection and analysis rule belongs.
- **Status** is the status of the scheduler. The following values are possible:
  - **Enabled** means the scheduler is running, and the data collection and analysis rule will be started in accordance with the specified schedule.
  - **Disabled** means the scheduler is not running.

    This is the default status of a newly created scheduler. For the scheduler to run, it must be **Enabled**.
- **The scheduler finished at** is the last time the scheduler's data collection and analysis rule was started.
- **Rule run status** is the status with which the scheduler has finished. The following values are possible:
  - **Ok** means the scheduler finished without errors, the rule was started.
  - **Unknown** means the scheduler was **Enabled** and its status is currently unknown. The **Unknown** status is displayed if you have linked storages and correlators on the corresponding tabs and **Enabled** the scheduler, but have not yet started it.
  - **Stopped** means the scheduler is stopped, the rule is not running.
  - **Error** means the scheduler has finished, and the rule was completed with an error.
- **Recent Error** lists errors (if any) that occurred during the execution of the data collection and analysis rule.

Failure to send events to the configured correlator does not constitute an error.

You can use the toolbar in the upper part of the table to **perform actions on schedulers**.

- Add a new scheduler. Click the **Add** button, and in the displayed window, select the check boxes next to the names of the data collection and analysis rules for which you want to create a scheduler.

  In this window, you can select only data collection and analysis rules that have been created previously. You cannot create a new rule.

- Remove a scheduler. In the table of schedulers, select the check boxes next to the schedulers that you want to delete and click the **Delete** button. The scheduler and links are removed. The data collection and analysis rule is not removed.

- Enable a scheduler. In the table of schedulers, select the check boxes next to the schedulers that you want to enable and click the **Enable according to schedule** button. The data collection and analysis rule for which this scheduler was created will be executed in accordance with the schedule configured in the settings of this rule.

- Disable a scheduler. In the table of schedulers, select the check boxes next to the schedulers that you want to disable and click the **Disable** button. The data collection and analysis rule is paused, but the scheduler itself and the links are not deleted.

- Start the scheduler. In the schedulers table, select the check boxes next to the enabled schedulers and click the **Run now** button. This scheduler's data collection and analysis rule is executed immediately.

To edit the scheduler, click the corresponding line in the table.

Available scheduler settings for data collection and analysis rules are described below.

General tab

On this tab you can:

- Enable or disable the scheduler using a toggle switch.

  If the toggle switch is enabled, the data collection and analysis rule runs in accordance with the schedule configured in its settings.

- Edit the following settings of the data collection and analysis rule:

  - Name
  - Query interval
  - Depth
  - Sql
  - Description
  - Mapping

The Linked storages tab

On this tab you need to specify the storage to which the scheduler will send SQL queries.

To specify a storage:

1. Click the **Link** button in the toolbar.
2. This opens a window; in that window, specify the name of the storage to which you want to add the link, as well as the name of the section of the selected storage.

   > You can select only one storage, but multiple sections of that storage.

3. Click **Add**.

   The link is created and displayed in the table on the **Storage bindings** tab.

If necessary, you can remove the links by selecting the check boxes in the relevant rows of the table and clicking the **Unbind selected** button.

The Linked correlators tab

On this tab, you must add correlators for handling base events.

To add a correlator:

1. Click the **Link** button in the toolbar.
2. This opens a window; in that window, hover over the **Correlator** field.
3. In the displayed list of correlators, select check boxes next to the correlators you want to add.
4. Click **Add**.

   The correlators are added and displayed in the table on the **Correlator bindings** tab.

If necessary, you can remove the correlators by selecting the check boxes in the relevant rows of the table and clicking the **Unbind selected** button.

You can also view the result of the scheduler in the Core log; to do so, you must first configure the **Debug** mode in Core settings. To download the log, select the **Resources** → **Active services** in KUMA, then select the Core service and click the **Log** button.

Log records with scheduler results have the `datamining scheduler` prefix.

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 is 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 is used to create correlation events if a certain event is found.

- operational is used for operations with active lists and context tables. This rule kind cannot create correlation events.

- periodical is 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 KUMA web interface.

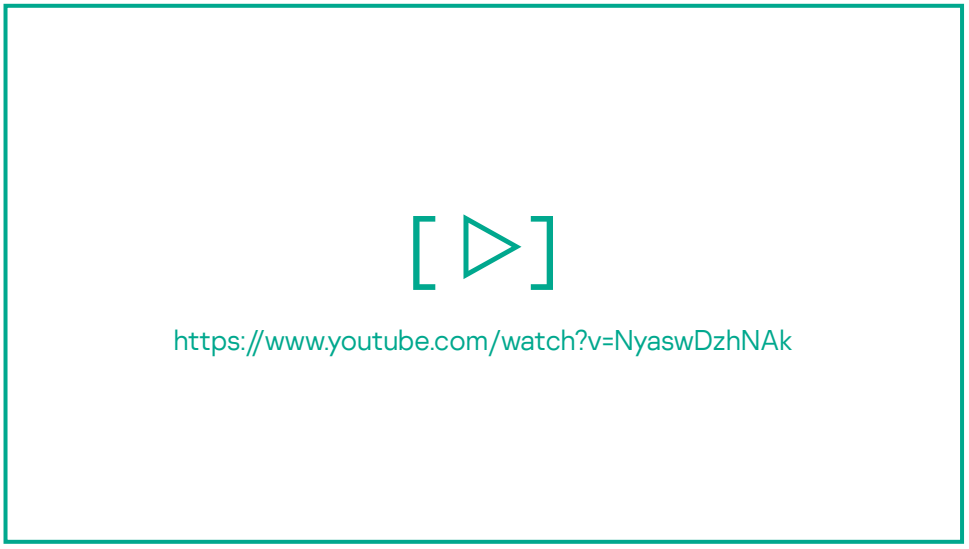
  When changing the correlator-ng configuration, you may need to delete the correlation/ and containers/ directories. Deleting these directories results in the loss of all accumulated correlator data, including statistics, correlation rule buckets, active lists, and context tables.

For these resources, you can enable the display of control characters in all input fields except the **Description** field.

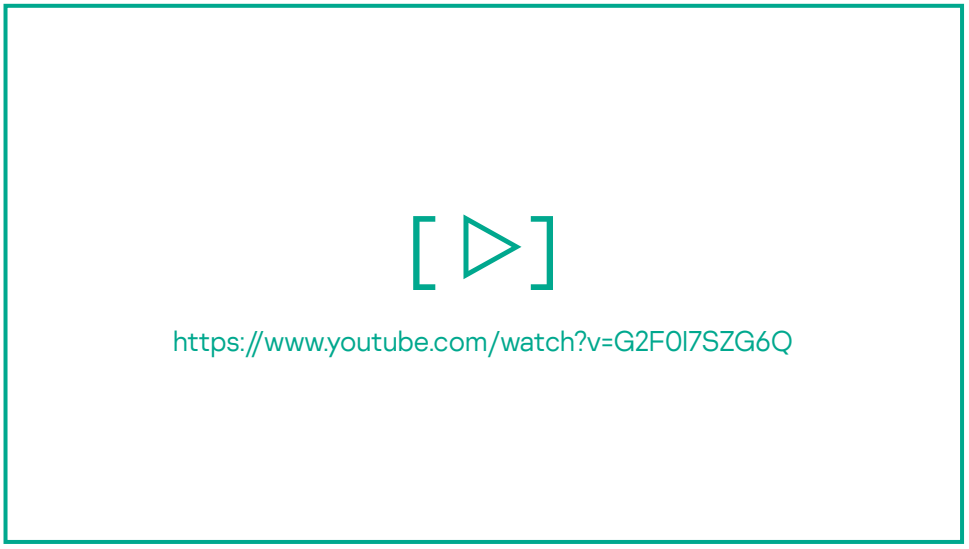
If a correlation rule is used in the correlator and an alert was created based on it, any change to the correlation rule will not result in a change to the existing alert even if the correlator service is restarted. For example, if the name of a correlation rule is changed, the name of the alert will 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.

If a correlation rule uses a different resource, for example, a previously saved filter, and this resource is modified, then if the correlation rule exported to a file may contain data from an earlier version of the modified

resource. However, subsequent import of the correlation rule from this file is completed correctly, and KUMA imports the latest version of the resource.



https://www.youtube.com/watch?v=NyaswDzhNAk



https://www.youtube.com/watch?v=G2F0l7SZG6Q

Correlation rules of the 'standard' type

Correlation rules of the **standard** type are used for identifying 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`.

In comparison with the data model of the normalized event, some fields of the correlation event formed by the **standard** correlation rule may be different (see the table below).

Name of the correlation event field	Description
StartTime	The earliest base event creation date and time (timestamp) among all base events from which the correlation event was generated.
EndTime	The later of the following dates and times: • Date and time when the correlation event was created. • The latest base event creation date and time (timestamp) among all base events from which the correlation event was generated.

To add fields from base events to the correlation event, specify these fields in the **Identical fields** and **Unique fields** parameters (see the table below).

Settings for a correlation rule of the **standard** type are described in the following tables.

General tab

This tab lets you specify the general settings of the correlation rule.

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.

Setting	Description
Type	Correlation rule type: **standard**. Required setting.
Tags	Tags for resource search. Optional setting.
Identical fields	Event fields that must be grouped in a Bucket. The hash of the values of the selected event fields is used as the Bucket key. If one of the selectors specified on the **Selectors** tab is triggered, the selected event fields are copied to the correlation event. If different selectors of the correlation rule use event fields that have different meanings in the events, do not specify such event fields in the **Identical fields** drop-down list. You can specify local variables. To refer to a local variable, its name must be preceded with the `$` character. For an example of using local variables, refer to the rule provided with KUMA: **R403_Access to malicious resources from a host with disabled protection or an out-of-date anti-virus database**. Required setting.
Window, sec	Bucket lifetime in seconds. The time starts counting when the bucket is created, when the bucket receives the first event. When the bucket lifetime expires, the trigger specified on the **Actions** → **On timeout** tab is triggered, and the container is deleted. Triggers specified on the **Actions** → **On every threshold** and **On subsequent thresholds** tabs can trigger more than once during the lifetime of the bucket. Required setting.
Unique fields	Unique event fields to be sent to the bucket. If you specify unique event fields, only these event fields will be sent to the container. The hash of values of the selected fields is used as the Bucket key. You can specify local variables. To refer to a local variable, its name must be preceded with the `$` character. 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**.
Rate limit	Maximum number of times a correlation rule can be triggered per second. The default value is `0`. If correlation rules employing complex logic for pattern detection are not triggered, this may be due to the way rule triggers are counted in KUMA. In this case, we recommend increasing the **Rate limit**, for example, to `1000000`.
Base events keep policy	This drop-down list lets you select base events that you want to put in the correlation event: • **first**—this option is used to store the first base event of the event collection that triggered creation of the correlation event. This value is selected by default. • **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.
Severity	Base coefficient used to determine the importance of a correlation rule: • **Critical** • **High** • **Medium** • **Low** (default)
Order by	Event field to be used by selectors of the correlation rule to track the evolution of the situation. This can be useful, for example, if you want to configure a correlation rule to be triggered when several types of events occur in a sequence.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.
MITRE techniques	Downloaded MITRE ATT&CK techniques for analyzing the security coverage status using the MITRE ATT&CK matrix.
Use Unique fields mapping	This toggle switch allows you to save the values of unique fields to an array and pass it to a correlation event field. If the toggle switch is enabled, in the lower part of the **General** tab, an additional **Unique fields mapping** group of settings is displayed, in which you can configure the mapping of the source original unique fields to correlation event fields. When processing an event using a correlation rule, field mapping takes place first, and then operations from the **Actions** tab are applied to the correlation event resulting from the initial mapping. The toggle switch is turned off by default. Optional setting.

Unique field mapping group of settings

If you need to pass values of fields listed under **Unique fields** to the correlation event, here you can configure the mapping of unique fields to correlation event fields. This group of settings is displayed on the **General** tab if the **Use Unique fields mapping** toggle switch is enabled. Values of unique fields are an array, therefore the field in the correlation event must have the appropriate type: SA, NA, FA.

You can add a mapping by clicking the **Add** button and selecting a field from the drop-down list in the **Source event field** column. You can select fields specified in the **Unique fields** parameter. In the drop-down list in the **Destination event field** column, select the correlation event field to which you want to write the array of values of the source field. You can select fields whose type matches the type of the array (SA, NA, or FA, depending on the type of the source field).

You can delete one or more mappings by selecting the check boxes next to the relevant mappings and clicking **Delete**.

Selectors tab

This tab is used to define the conditions that the processed events must fulfill to trigger the correlation rule. To add a selector, click the **+ Add selector** button. You can add multiple selectors, reorder selectors, or remove selectors. To reorder selectors, use the reorder ⣿ icons. To remove a selector, click the delete ✕ icon next to it.

Each selector has a **Settings** tab and a **Local variables** tab.

The settings available on the **Settings** tab are described in the table below.

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Selector threshold (event count)	The number of events that must be received for the selector to trigger. The default value is `1`. Required setting.
Recovery	This toggle switch lets the correlation rule not trigger when the selector receives the number of events specified in the **Selector threshold (event count)** field. This toggle switch is turned off by default. **Example of how a selector works with the Recovery toggle switch enabled** The **Recovery** switch is used in selectors of correlation rules to reset the bucket and prevent a correlation event from being generated. Typically, recovery is used together with the **On timeout** action. The **standard** correlation rule is configured as follows: • **Window, sec** is set to `300 seconds`. • On the **Actions** tab, the **On timeout** trigger is configured with the **Output** check box selected. • Two selectors have been added to the correlation rule: • **Virus detected** (in the example, the `{"Message": "virus detected"}` event) with the **Recovery** toggle switch disabled. • **Virus deleted** (in the example, the `{"Message": "virus deleted"}` event) with the **Recovery** toggle switch enabled. When a `{"Message": "virus detected"}` event is received, the bucket is started and the following happens: • A `{"Message": "virus deleted"}` event is not received within 300 seconds. In this case, the bucket is closed and the trigger generates a correlation event. • Within 300 seconds, a `{"Message": "virus deleted"}` event is received, which matches a selector with the **Recovery** toggle switch enabled. In this case, the bucket is cleared and no correlation event is generated.

Setting	Description
Filter	The filter that defines criteria for identifying events that trigger the selector when received. 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

Setting	Description

	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.
	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 specified in selector filter 2 is preferable because it places less load on the system.

On the **Local variables** tab, you can add variables that will be valid inside the correlation rule. To add a variable, click the **+ Add** button, then specify the variable and its value. You can add multiple variables or delete variables. To delete a variable, select the check box next to it and click the **Delete** button.

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. 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` 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 the syntax and examples of correlation rules that use regular expressions in their selectors, refer to 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

You can use this tab to configure the triggers of the correlation rule. You can configure triggers on the following tabs:

- **On first threshold** triggers when the Bucket registers the first triggering of the selector during the lifetime of the Bucket.
- **On subsequent thresholds** triggers when the Bucket registers the second and all subsequent triggering of the selector during the lifetime of the Bucket.
- **On every threshold** triggers every time the Bucket registers the triggering of the selector.
- **On timeout** triggers when the lifetime of the Bucket ends, and is used together with a selector that has the **Recovery** check box selected in its settings. Thus, this trigger activates if the situation detected by the correlation rule is not resolved within the specified lifetime.

Available trigger settings are listed in the table below.

Setting	Description

Enrichment

Enrichment rules for modifying the values of correlation event fields. Enrichment rules are stored in the correlation rule where they were created. To create an enrichment rule, click the **+ Add enrichment** button.

Available enrichment rule settings:

- **Original type** is the type of the enrichment. When you select some enrichment types, additional settings may become available that you must specify.
  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 enrichment fields** setting 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 extended schema array 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. 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.
Mapping	Event fields for data transfer: • **Dictionary field** specifies dictionary fields from which data is to be transmitted. The available fields depend on the selected dictionary resource. • **KUMA field** specifies event fields to which data is to be transmitted. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written there.

    The first field in the table (**Dictionary field**) is taken as the key with which the fields selected from the event as key fields are matched (**KUMA field**). As the key in the **Dictionary field**, you must select an indicator of compromise by which the enrichment is to be performed, for example, IP address, URL, or hash. In the rule, you must select the event field that corresponds to the selected indicator of compromise in the dictionary field.

    If you want to select multiple key fields, you can specify them using `|` as a separator (when specifying in the web interface or importing as a CSV file), for example, `< IP address >|< user name >`.

    You can add new table rows or delete table rows. To add a new table row, click **Add new element**. To delete a row in the table, click 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 for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, 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.
  - Converting the IP address to the IPv4 format:
    - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.
    - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

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**.

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.

  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 the result of 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}}
  ```

  Required setting.
- The **Debug** toggle switch enables resource logging. This toggle switch is turned off by default.
- **Tags**

You can create multiple enrichment rules, reorder enrichment rules, or delete enrichment rules. To reorder enrichment rules, use the reorder ⣿ icons. To delete an enrichment rule, click the delete ✕ icon next to it.

Output	This check box enables the sending of correlation events for *post-processing*, that is, for external enrichment outside the correlation rule, for response, and to destinations. By default, this check box is cleared.
Loop to correlator	This check box enables the processing of the created correlation event by the rule chain of the current correlator. This makes hierarchical correlation possible. By default, this check box is cleared. If the **Output** and **Loop to correlator** check boxes are selected, the correlation rule is sent to post-processing first, and then to the selectors of the current correlation rule.
  ```

No alert	The check box disables the creation of alerts when the correlation rule is triggered. By default, this check box is cleared. 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.
Categorization	Categorization rules for assets involved in the event. Using categorization rules, you can link and unlink only reactive categories to and from assets. To create an enrichment rule, click the **+ Add categorization** button. Available categorization rule settings: • **Action** is the operation applied to the category: • **Add**: Link the category to the asset. • **Delete**: Unlink the category from the asset. Required setting. • **Event field** is the field of the event that contains the asset to which the operation will be applied. Required setting. • **Category ID** is the category to which the operation will be applied. Required setting. You can create multiple categorization rules, reorder categorization rules, or delete categorization rules. To reorder categorization rules, use the reorder ⣿ icons. To delete a categorization rule, click the delete ✕ icon next to it.
Active lists update	Operations with active lists. To create an operation with an active list, click the **+ Add active list action** button. Available parameters of an active list operation: • **Name** specifies the active list to which the operation is applied. If you want to edit the settings of an active list, click the pencil icon next to it. Required setting. • **Operation** is the operation that is applied to the active list: • **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. • **Get**—get the Active list entry and write the values of the selected fields into the correlation event. • **Delete**—delete the Active list entry. Required setting. • **Key fields** are event fields that are used to create an active list entry. The specified event fields are also used as the key of the active list entry. The active list entry key depends on the available event fields and does not depend on the order in which they are displayed in the KUMA web interface. Required setting. • **Mapping**: Rules for mapping active list fields to event fields. You can use mapping rules if in the **Operation** drop-down list, you selected **Get** or **Set**. To create a mapping rule, click the **+ Add** button. Available mapping rule settings: • **Active list field** is the active list field that is mapped to the event field. The field must not contain special characters or numbers only. • **KUMA field** is the event field to which the active list field is mapped. • **Constant** is a constant that is assigned to the active list field. You need to specify a constant if in the **Operation** drop-down list, you selected **Set**. You can create multiple mapping rules or delete mapping rules. To delete a mapping rule, select the check box next to it and click **Delete**. You can create multiple operations with active lists, reorder operations with active lists, or delete operations with active lists. To reorder operations with active lists, use the reorder ⣿ icons. To delete an operation with an active list, click the delete ✕ icon next to it.

Context tables update	Operations with context tables. To create an operation with a context table, click the **+ Add context table action** button. Available parameters of a context table operation: • **Name** specifies the context table to which the operation is applied. If you want to edit the settings of a context table, click the pencil icon next to it. Required setting. • **Operation** is the operation that is applied to the context table: • **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. • **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. • **Delete**—delete the context table entry. Required setting. • **Mapping**: Rules for mapping context table fields to event fields or variables. You can use mapping rules if in the **Operation** drop-down list, you selected something other than **Delete**. To create a mapping rule, click the **+ Add** button. Available mapping rule settings: • **Context table field** is the context table field that is mapped to an event field. You cannot specify a context table field that is already used in a mapping. You can specify tabulation characters, special characters, or just numbers. The maximum length of a context table field name is 128 characters. A context table field name cannot begin with an underscore. • **KUMA field** is the event field or local variable to which the context table field is mapped. • **Constant** is a constant that is assigned to the context table field. You need to specify a constant if in the **Operation** drop-down list, you selected **Set**, **Merge**, or **Sum**. The maximum length of a constant is 1024 characters. You can create multiple mapping rules or delete mapping rules. To delete a mapping rule, select the check box next to it and click **Delete**. You can create multiple operations with context tables, reorder operations with context tables, or delete operations with context tables. To reorder operations with context tables, use the reorder icons. To delete an operation with a context tables, click the delete icon next to it.

Correlators tab

This tab is displayed only when you edit the settings of the created correlation rule; on this tab, you can link the correlation rule to a correlator.

To link the correlation rule to a correlator, click **Link to service → Link to correlator** button, in the displayed window, specify one or more correlators, and click the **OK** button. The correlation rule is linked to the specified correlators and added to the end of the execution queue in the correlator settings. If you want to change the position of a correlation rule in the execution queue, go to the **Resources → Correlator** section, click the correlator, and in the displayed window, go to the **Correlation** section, select the check box next to the correlation rule, and change the position of the correlation rule by clicking the **Move up** and **Move down** buttons.

You can add multiple correlators or delete correlators. To do this, select the check box next to the correlator, click the **Delete** button, and confirm deletion.

Correlation rules of the 'simple' type

Correlation rules of the **simple** type are used to define simple sequences of events. Settings for a correlation rule of the **simple** type are described in the following tables.

In comparison with the data model of the normalized event, some fields of the correlation event formed by the **simple** correlation rule may be different (see the table below).

Name of the correlation event field	Description
StartTime	Creation date and time (timestamp) of the base event from which the correlation event was generated.
EndTime	Creation date and time (timestamp) of the base event from which the correlation event was generated.

General tab

This tab lets you specify the general settings of the correlation rule.

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	Correlation rule type: **simple**. Required setting.
Tags	Tags for resource search. Optional setting.
Propagated fields	Event fields by which events are selected. If a selector specified on the **Selectors** tab is triggered, the selected event fields are copied to the correlation event.
Rate limit	Maximum number of times a correlation rule can be triggered per second. The default value is `0`. If correlation rules employing complex logic for pattern detection are not triggered, this may be due to the way rule triggers are counted in KUMA. In this case, we recommend increasing the **Rate limit**, for example, to `1000000`.
Severity	Base coefficient used to determine the importance of a correlation rule: • **Critical** • **High** • **Medium** • **Low** (default)
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.
MITRE techniques	Downloaded MITRE ATT&CK techniques for analyzing the security coverage status using the MITRE ATT&CK matrix.

Selectors tab

This tab is used to define the conditions that the processed events must fulfill to trigger the correlation rule. A selector has a **Settings** tab and a **Local variables** tab.

The settings available on the **Settings** tab are described in the table below.

Setting	Description
Filter	The filter that defines criteria for identifying events that trigger the selector when received. 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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**.

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.
	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 specified in selector filter 2 is preferable because it places less load on the system.

On the **Local variables** tab, you can add variables that will be valid inside the correlation rule. To add a variable, click the **+ Add** button, then specify the variable and its value. You can add multiple variables or delete variables. To delete a variable, select the check box next to it and click the **Delete** button.

Actions tab

You can use this tab to configure the trigger of the correlation rule. A correlation rule of the **simple** type can have only one trigger, which is activated each time the bucket registers the selector triggering. Available trigger settings are listed in the table below.

Setting	Description

Enrichment

Enrichment rules for modifying the values of correlation event fields. Enrichment rules are stored in the correlation rule where they were created. To create an enrichment rule, click the **+ Add enrichment** button.

Available enrichment rule settings:

- **Original type** is the type of the enrichment. When you select some enrichment types, additional settings may become available that you must specify.
  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 enrichment fields** setting 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 extended schema array 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. 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.
Mapping	Event fields for data transfer: • **Dictionary field** specifies dictionary fields from which data is to be transmitted. The available fields depend on the selected dictionary resource. • **KUMA field** specifies event fields to which data is to be transmitted. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written there.

    The first field in the table (**Dictionary field**) is taken as the key with which the fields selected from the event as key fields are matched (**KUMA field**). As the key in the **Dictionary field**, you must select an indicator of compromise by which the enrichment is to be performed, for example, IP address, URL, or hash. In the rule, you must select the event field that corresponds to the selected indicator of compromise in the dictionary field.

    If you want to select multiple key fields, you can specify them using `|` as a separator (when specifying in the web interface or importing as a CSV file), for example, `< IP address >|< user name >`.

    You can add new table rows or delete table rows. To add a new table row, click **Add new element**. To delete a row in the table, click 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 for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, 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.
  - Converting the IP address to the IPv4 format:
    - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.
    - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

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**.

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.

  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 the result of 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}}
  ```

  Required setting.
- The **Debug** toggle switch enables resource logging. This toggle switch is turned off by default.
- **Tags**

You can create multiple enrichment rules, reorder enrichment rules, or delete enrichment rules. To reorder enrichment rules, use the reorder ⣿ icons. To delete an enrichment rule, click the delete ✕ icon next to it.

Output	This check box enables the sending of correlation events for *post-processing*, that is, for external enrichment outside the correlation rule, for response, and to destinations. By default, this check box is cleared.
Loop to correlator	This check box enables the processing of the created correlation event by the rule chain of the current correlator. This makes hierarchical correlation possible. By default, this check box is cleared. If the **Output** and **Loop to correlator** check boxes are selected, the correlation rule is sent to post-processing first, and then to the selectors of the current correlation rule.
  ```

No alert	The check box disables the creation of alerts when the correlation rule is triggered. By default, this check box is cleared. 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.
Categorization	Categorization rules for assets involved in the event. Using categorization rules, you can link and unlink only reactive categories to and from assets. To create an enrichment rule, click the **+ Add categorization** button. Available categorization rule settings: • **Action** is the operation applied to the category: • **Add**: Link the category to the asset. • **Delete**: Unlink the category from the asset. Required setting. • **Event field** is the field of the event that contains the asset to which the operation will be applied. Required setting. • **Category ID** is the category to which the operation will be applied. Required setting. You can create multiple categorization rules, reorder categorization rules, or delete categorization rules. To reorder categorization rules, use the reorder ⣿ icons. To delete a categorization rule, click the delete ✕ icon next to it.
Active lists update	Operations with active lists. To create an operation with an active list, click the **+ Add active list action** button. Available parameters of an active list operation: • **Name** specifies the active list to which the operation is applied. If you want to edit the settings of an active list, click the pencil 🖉 icon next to it. Required setting. • **Operation** is the operation that is applied to the active list: • **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. • **Get**—get the Active list entry and write the values of the selected fields into the correlation event. • **Delete**—delete the Active list entry. Required setting. • **Key fields** are event fields that are used to create an active list entry. The specified event fields are also used as the key of the active list entry. > The active list entry key depends on the available event fields and does not depend on the order in which they are displayed in the KUMA web interface. Required setting. • **Mapping**: Rules for mapping active list fields to event fields. You can use mapping rules if in the **Operation** drop-down list, you selected **Get** or **Set**. To create a mapping rule, click the **+ Add** button. Available mapping rule settings: • **Active list field** is the active list field that is mapped to the event field. The field must not contain special characters or numbers only. • **KUMA field** is the event field to which the active list field is mapped. • **Constant** is a constant that is assigned to the active list field. You need to specify a constant if in the **Operation** drop-down list, you selected **Set**. You can create multiple mapping rules or delete mapping rules. To delete a mapping rule, select the check box next to it and click **Delete**. You can create multiple operations with active lists, reorder operations with active lists, or delete operations with active lists. To reorder operations with active lists, use the reorder ⣿ icons. To delete an operation with an active list, click the delete ✕ icon next to it.

Context tables update	Operations with context tables. To create an operation with a context table, click the **+ Add context table action** button. Available parameters of a context table operation: • **Name** specifies the context table to which the operation is applied. If you want to edit the settings of a context table, click the pencil icon next to it. Required setting. • **Operation** is the operation that is applied to the context table: • **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. • **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. • **Delete**—delete the context table entry. Required setting. • **Mapping**: Rules for mapping context table fields to event fields or variables. You can use mapping rules if in the **Operation** drop-down list, you selected something other than **Delete**. To create a mapping rule, click the **+ Add** button. Available mapping rule settings: • **Context table field** is the context table field that is mapped to an event field. You cannot specify a context table field that is already used in a mapping. You can specify tabulation characters, special characters, or just numbers. The maximum length of a context table field name is 128 characters. A context table field name cannot begin with an underscore. • **KUMA field** is the event field or local variable to which the context table field is mapped. • **Constant** is a constant that is assigned to the context table field. You need to specify a constant if in the **Operation** drop-down list, you selected **Set**, **Merge**, or **Sum**. The maximum length of a constant is 1024 characters. You can create multiple mapping rules or delete mapping rules. To delete a mapping rule, select the check box next to it and click **Delete**. You can create multiple operations with context tables, reorder operations with context tables, or delete operations with context tables. To reorder operations with context tables, use the reorder icons. To delete an operation with a context tables, click the delete icon next to it.

Correlators tab

This tab is displayed only when you edit the settings of the created correlation rule; on this tab, you can link the correlation rule to a correlator.

To link the correlation rule to a correlator, click **Link to service → Link to correlator** button, in the displayed window, specify one or more correlators, and click the **OK** button. The correlation rule is linked to the specified correlators and added to the end of the execution queue in the correlator settings. If you want to change the position of a correlation rule in the execution queue, go to the **Resources → Correlator** section, click the correlator, and in the displayed window, go to the **Correlation** section, select the check box next to the correlation rule, and change the position of the correlation rule by clicking the **Move up** and **Move down** buttons.

You can add multiple correlators or delete correlators. To do this, select the check box next to the correlator, click the **Delete** button, and confirm deletion.

Correlation rules of the 'operational' type

Correlation rules of the **operational** type are used for working with active lists. Settings for a correlation rule of the **operational** type are described in the following tables.

General tab

This tab lets you specify the general settings of the correlation rule.

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	Correlation rule type: **operational**. Required setting.
Tags	Tags for resource search. Optional setting.
Rate limit	Maximum number of times a correlation rule can be triggered per second. The default value is `0`. If correlation rules employing complex logic for pattern detection are not triggered, this may be due to the way rule triggers are counted in KUMA. In this case, we recommend increasing the **Rate limit**, for example, to `1000000`.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.
MITRE techniques	Downloaded MITRE ATT&CK techniques for analyzing the security coverage status using the MITRE ATT&CK matrix.

Selectors tab

This tab is used to define the conditions that the processed events must fulfill to trigger the correlation rule. A selector has a **Settings** tab and a **Local variables** tab.

The settings available on the **Settings** tab are described in the table below.

Setting	Description
Filter	The filter that defines criteria for identifying events that trigger the selector when received. 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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**.

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.
	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 specified in selector filter 2 is preferable because it places less load on the system.

On the **Local variables** tab, you can add variables that will be valid inside the correlation rule. To add a variable, click the **+ Add** button, then specify the variable and its value. You can add multiple variables or delete variables. To delete a variable, select the check box next to it and click the **Delete** button.

Actions tab

You can use this tab to configure the trigger of the correlation rule. A correlation rule of the **operational** type can have only one trigger, which is activated each time the bucket registers the selector triggering. Available trigger settings are listed in the table below.

Setting	Description
Active lists update	Operations with active lists. To create an operation with an active list, click the **+ Add active list action** button. Available parameters of an active list operation: • **Name** specifies the active list to which the operation is applied. If you want to edit the settings of an active list, click the pencil icon next to it. Required setting. • **Operation** is the operation that is applied to the active list: • **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. • **Get**—get the Active list entry and write the values of the selected fields into the correlation event. • **Delete**—delete the Active list entry. Required setting. • **Key fields** are event fields that are used to create an active list entry. The specified event fields are also used as the key of the active list entry. The active list entry key depends on the available event fields and does not depend on the order in which they are displayed in the KUMA web interface. Required setting. • **Mapping**: Rules for mapping active list fields to event fields. You can use mapping rules if in the **Operation** drop-down list, you selected **Get** or **Set**. To create a mapping rule, click the **+ Add** button. Available mapping rule settings: • **Active list field** is the active list field that is mapped to the event field. The field must not contain special characters or numbers only. • **KUMA field** is the event field to which the active list field is mapped. • **Constant** is a constant that is assigned to the active list field. You need to specify a constant if in the **Operation** drop-down list, you selected **Set**. You can create multiple mapping rules or delete mapping rules. To delete a mapping rule, select the check box next to it and click **Delete**. You can create multiple operations with active lists, reorder operations with active lists, or delete operations with active lists. To reorder operations with active lists, use the reorder icons. To delete an operation with an active list, click the delete icon next to it.

Setting	Description
Context tables update	Operations with context tables. To create an operation with a context table, click the **+ Add context table action** button. Available parameters of a context table operation: • **Name** specifies the context table to which the operation is applied. If you want to edit the settings of a context table, click the pencil icon next to it. Required setting. • **Operation** is the operation that is applied to the context table: • **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. • **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. • **Delete**—delete the context table entry. Required setting. • **Mapping**: Rules for mapping context table fields to event fields or variables. You can use mapping rules if in the **Operation** drop-down list, you selected something other than **Delete**. To create a mapping rule, click the **+ Add** button. Available mapping rule settings: • **Context table field** is the context table field that is mapped to an event field. You cannot specify a context table field that is already used in a mapping. You can specify tabulation characters, special characters, or just numbers. The maximum length of a context table field name is 128 characters. A context table field name cannot begin with an underscore. • **KUMA field** is the event field or local variable to which the context table field is mapped. • **Constant** is a constant that is assigned to the context table field. You need to specify a constant if in the **Operation** drop-down list, you selected **Set**, **Merge**, or **Sum**. The maximum length of a constant is 1024 characters. You can create multiple mapping rules or delete mapping rules. To delete a mapping rule, select the check box next to it and click **Delete**. You can create multiple operations with context tables, reorder operations with context tables, or delete operations with context tables. To reorder operations with context tables, use the reorder icons. To delete an operation with a context tables, click the delete icon next to it.

Correlators tab

This tab is displayed only when you edit the settings of the created correlation rule; on this tab, you can link the correlation rule to a correlator.

To link the correlation rule to a correlator, click **Link to service → Link to correlator** button, in the displayed window, specify one or more correlators, and click the **OK** button. The correlation rule is linked to the specified correlators and added to the end of the execution queue in the correlator settings. If you want to change the position of a correlation rule in the execution queue, go to the **Resources → Correlator** section, click the correlator, and in the displayed window, go to the **Correlation** section, select the check box next to the correlation rule, and change the position of the correlation rule by clicking the **Move up** and **Move down** buttons.

You can add multiple correlators or delete correlators. To do this, select the check box next to the correlator, click the **Delete** button, and confirm deletion.

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.

   How to start a retroscan

   You can start a retroscan only from the command line; you cannot do this through the KUMA web interface.

   To run a retroscan:

   1. Create a query to download events.

      You need to prepare a query to the user authentication event storage. The data must be sorted by event creation time in ascending order.

      Query example:

      ```
      SELECT *
      FROM `events`
      WHERE SourceUserID == 's-1-5-21-1445412355-99643495-187345912-1119' AND Type =
      1
      ORDER BY Timestamp ASC
      LIMIT 250
      ```

      For event IDs to be displayed correctly in the correlator logs, we recommend including the event ID field when exporting data.

   2. Export the query result to a TSV file.

      Events in the file must be sorted by creation time.

3. Start the retroscan on the command line by running the following command:

```
/opt/kaspersky/kuma/kuma tools retroscan \
  --url <KUMA Core URL>:<KUMA Core port> \
  --src <path to the exported TSV file with events> \
  --cert <path to the KUMA Core certificate> \
  --key <path to the key for connecting to the KUMA Core> \
  --alerts
```

The `--alerts` option must be specified if you want to create an alert.

Downloaded events are processed by the correlation rule in the same way as events coming from the stream. If the downloaded data covers the minimum cold start period, periodic anomaly detection starts immediately after the download is complete.

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 chosing 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.

For more details on using correlator-ng, see the Creating a correlator-ng correlator section

*To create a correlation rule of the **periodical** type:*

1. Create a correlator service in the KUMA web interface.

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 (unique user IDs) 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`.

Setting	Description
`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 maximum value of this setting depends on the `eventGroupingPeriod` and `countersCollectingPeriod` settings and is determined by the number of data grouping periods available for analysis. The maximum value can be calculated using the following formula: `minSamples` ≤ `countersCollectingPeriod` / `eventGroupingPeriod`. The minimum value is `15`, the recommended value is `15`. The maximum value is `60`.
`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 maximum value of this setting depends on the `eventGroupingPeriod` setting and is calculated using the formula: `detectionPeriod` ≤ `eventGroupingPeriod` * `1440`. 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. This setting affects the maximum allowed `detectionPeriod` value. The grouped data is used for analysis. 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 `60`, 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 maximum value can be calculated using the following formula: `countersCollectingPeriod` = number of periods * `eventGroupingPeriod`. The minimum value can be calculated using the following formula: `15` * `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 maximum value can be calculated using the following formula: `retentionPeriod` = number of periods * `eventGroupingPeriod`. The minimum value can be calculated using the following formula: `15` * `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.
`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.

You can use extended event schema fields in correlation rules, local variables, and global variables.

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

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. Active list name.
2. Expression that returns the field name of the active list
3. One or more expressions whose results make up 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.

You can use these functions to reference an 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

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 make up the dictionary string 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.

You can use this function to reference a 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

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.

You can use this function to reference a 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

Returns the value of the specified field from the context table.

You must specify the parameters in the following sequence:

1. Name of the context table. The name must be specified.
2. Name of the field whose value you want to get.
3. Name of the key field to be compared to.
4. Expression that returns the value for comparison with the value of the key field from item 3.

The function must contain at least 4 parameters.

Usage example	Result
`context_table('test_table', 'Username', 'IP', DestinationAddress)`	Getting the value of the specified field from a row of the context table for which the key field matches the value passed in the comparison parameter. If there is no match, the table does not exist, or the field is not found, an empty string is returned.

You can use this function to reference a context table of the Shared tenant from a variable. To do so, add the `@Shared` suffix after the name of the context table (case sensitive). For example, `context_table('test_table@Shared', SourceAddress)`.

len

Returns the length of a string or array.

The function returns the length of the array if the passed array belongs to 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

Returns a list of unique elements in an array.

The function returns the list of unique array elements if the passed array belongs to 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

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 sorted list of array elements if the passed array belongs to 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

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

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

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

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

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

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. Substring end position (optional).

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

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 a 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

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 -922337203685477580.

The function accepts the following parameters:

- As source data, an event field, another variable, or a 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

Deletes 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. Substring to be deleted from the beginning and end of the original string (optional)

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

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

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

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

Returns the string specified in the function, with variables replaced with their values. The template function can be used in one of two ways:

- With a single argument. In this case, you can use the field names to substitute values in the template.
- With multiple arguments. In this case, the zero-based index of the argument is used to substitute values.

Usage examples
• `template('Message text: {{.Message}}')` • `template('Name in lower case: {{index . 0}}. Name in upper case: {{index . 1}}', to_lower(Name), to_upper(Name))`

Operations with timestamps

now function

Gets a timestamp in epoch format. Runs with no arguments.

Usage examples
`now()`

extract_from_timestamp

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. 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')`

parse_timestamp

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

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. Time zone designation (optional). 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

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. Time zone designation (optional). 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

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 extended event schema fields of the `integer` and `float` types.

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 string 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

Rounds numbers. Supported for extended event schema fields of the `integer` and `float` types.

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

Rounds up numbers. Supported for extended event schema fields of the `integer` and `float` types.

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

Rounds down numbers. Supported for extended event schema fields of the `integer` and `float` types.

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

Gets the modulus of a number. Supported for extended event schema fields of the `integer` and `float` types.

Available arguments:

- Numeric event fields
- Numeric variables
- Numeric constants

Usage examples (DeviceCustomNumber1=-7; otherVariable=-2)	Usage result
abs(DeviceCustomFloatingPoint1)	7
abs($otherVariable)	2

pow

Exponentiates a number. Supported for extended event schema fields of the `integer` and `float` types.

The parameters must be specified in the following sequence:

1. Base—real numbers.
2. Exponent—natural numbers or real numbers.

Available arguments:

- Numeric event fields
- Numeric variables
- Numeric constants

Usage examples
`pow(DeviceCustomNumber1, DeviceCustomNumber2)`
`pow($otherVariable, DeviceCustomNumber1)`

str_join

Join multiple strings into one using a separator. Supported for extended event schema fields of the `integer` and `float` types.

The parameters must be specified in the following sequence:

1. Separator. String.
2. String 1, string 2, string N. At least 2 expressions.

Usage examples	Usage result
`str_join('|', to_lower(Name), to_upper(Name), Name)`	String.

conditional

Get one value if a condition is met and another value if the condition is not met. Supported for extended event schema fields of the `integer` and `float` types.

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` (passes a RE2 regular expression, not an SQL expression)
- `ILIKE` (passes a RE2 regular expression, not 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 operations are supported:

- `len`
- `to_lower`
- `to_upper`
- `append`
- `prepend`
- `substring`
- `tr`
- `replace`
- `regexp_replace`
- `regexp_capture`

For extended event schema fields of the `integer` or `float` type, the following operations are supported:

- Basic mathematical operations:
  - `round`
  - `ceil`
  - `floor`
  - `abs`
  - `pow`
  - `str_join`
  - `conditional`

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"]`.
- Gett 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 KUMA web interface, under **Resources** → **Correlators**, select the resource set of the relevant correlator.

   The Correlator Installation Wizard opens.

2. Select the **Global variables** step of the Installation Wizard.

3. 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.

     When entering functions, you can use autocomplete as a list of hints with possible function names, their brief description and usage examples. You can select a function from the list and insert it together with its list of arguments into the input field.

     To display the list of all hints in the field, press **`Ctrl+Space`**. Press **`Enter`** to select a function from the list. Press **`Tab`** to go to the next argument in the list of arguments of the selected function.

     Description of variable functions.

   Multiple variables can be added. Added variables can be edited or deleted by using the ✕ icon.

4. 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 KUMA web interface, under **Resources** → **Correlation rules**, 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.

2. Click the **Selectors** tab.

3. 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.

  When entering functions, you can use autocomplete as a list of hints with possible function names, their brief description and usage examples. You can select a function from the list and insert it together with its list of arguments into the input field.

  To display the list of all hints in the field, press **`Ctrl+Space`**. Press **`Enter`** to select a function from the list. Press **`Tab`** to go to the next argument in the list of arguments of the selected 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.

4. 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.

Adding a temporary exclusion list for a correlation rule

Users that do not have the rights to edit correlation rules in the KUMA web interface, can create a temporary list of exclusions (for example, create exclusions for false positives when managing alerts). A user with the rights to edit correlation rules can then add the exclusions to the rule and remove them from the temporary list.

To add exclusions to a correlation rule when managing alerts:

1. Go to the **Alerts** section and select an alert.
2. Click the **Find in events** button.

   Events of the alert are displayed on the events page.
3. Open the correlation event.

   This opens the event card in which each field has a ⌄ (arrow) button that lets you add an exclusion.
4. Click the ⌄ button and select **Add to exclusions** .

   A sidebar is displayed, containing the following fields: **Correlation rule**, **Exclusion**, **Alert**, **Comment**.
5. Click the **Create** button.

The exclusion rule is added.

The exclusion is added to the temporary list. This list is available to anyone with rights to read correlation rules: in the **Resources** → **Correlation rules** section, in the toolbar of the rule list, click the **List of exclusions** button. If you want to view the exclusions of a specific rule, open the card of the rule and select the **Exclusions** tab.

The exclusion list contains entries with the following parameters:

- `Exclusion`

  Exclusion condition.

- `Correlation rule`

  Name of the correlation rule.

- `Alert`

  Name of the alert from which the exclusion was added.

- `Tenant`

  The tenant to which the rule and the exclusion apply.

- `Condition`

  Generated automatically based on the selected field of the correlation event.

- `Creation date`

  Date and time when the exclusion was added.

- `Expires`

Date and time when the exclusion will be automatically removed from the list.

- `Created`

  Name of the user that added the exclusion.

- `Comment`

After the exclusion is added, by default, the correlation rule takes the exclusion into account for 7 days. In the **Settings** → **Other** → **General** section, you can configure the duration of exclusions by editing the `corr_rule_exclusion_ttl_hours` parameter in the **Core properties** section. You can configure the lifetime of exclusions in hours and days. The minimum value is 1 hour, the maximum is 365 days. This setting is available only for users with the General administrator role.

For fields from base events to be propagated to correlation events, these fields must be specified in the card of the correlation rule on the **General** tab, in the **Propagated fields** field. If the fields of base events are not mapped to the correlation event, these fields cannot be added to exclusions.

To remove exclusions from a correlation rule:

1. Go to the **Resources** → **Correlation rules** section.
2. In the toolbar of the rule list, click **List of exclusions** button.

   This opens the window with the list of exclusions.
3. Select the exclusions that you want to delete and click the **Delete** button.

Exceptions are deleted from the correlation rule.

KUMA generates an audit event whenever an exception is created or deleted. You can view the changes of event settings in the **Event details** window.

Predefined correlation rules

The KUMA 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 of time	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] 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] 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 of time	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.

MITRE ATT&CK techniques and tactics

KUMA can:

- Enrich correlation events with information about MITRE ATT&CK techniques and tactics.

  Tactic and Technique fields of the event data model are used for this purpose. When generating a correlation event, these fields can be populated with relevant data for later use. For example, when a new alert is received with MITRE ATT&CK markup, you can open the MITRE ATT&CK website and read about the techniques and tactics to learn when, how, and why attackers might use these techniques, how to detect them, and how to mitigate risks — all of this can help you develop a response plan. You can also build reports and dashboards based on alerts and techniques detected in the infrastructure. If you are using correlation rules from SOC_package and want to customize the enrichment of correlation events with information about MITRE ATT&CK techniques and tactics, add the MITRE enrichment rules from SOC_package to the correlator.

- Assess the coverage of the MITRE ATT&CK matrix by your correlation rules.

  In this case, the general correlation rule parameters are used, which allow associating MITRE techniques with each rule. This parameter is used to describe the rule itself and this data is not passed to the correlation rule or alert in any way. Associating techniques and tactics with correlation rules lets you analyze the MITRE ATT&CK matrix coverage, focusing on the most relevant techniques for your specific infrastructure.

  If you want to assess the coverage of the MITRE ATT&CK matrix by your correlation rules:

  1. Download the list of techniques from the official MITRE ATT&CK repository and import it into KUMA.
  2. Map MITRE ATT&CK 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.

- Visualization of the MITRE ATT&CK matrix coverage by Kaspersky products.

Adding MITRE enrichment rules to the correlator

To add MITRE enrichment rules from SOC_package:

1. In the **Resources and services** → **Correlators** section in KUMA, click **Add** to create a correlator or select an existing correlator from the list to enter the settings editing mode.
2. In the Correlator Installation Wizard, on the **Enrichment** step, click **Add enrichment**.
3. This opens the **Enrichment** window; in that window, in the **Enrichment rule** drop-down list, select the **MITRE Technique** and **MITRE Tactics** rules.

   The **MITRE Technique** and **MITRE Tactics** enrichment rules are pre-configured, you do not need to specify any additional settings.
4. Save the correlator settings.
5. If you have created a correlator, you must install the correlator service on the host. If you edited the settings of an active correlator, you mustrestart the correlator service.

MITRE Technique and **MITRE Tactics** enrichment rules are added.

Importing the list of MITRE ATT&CK techniques

Only a user with the General Administrator role can import the list of MITRE ATT&CK techniques.

To import the list of MITRE ATT&CK techniques:

1. Download the list of MITRE ATT&CK techniques from the GitHub portal.

   KUMA 4.2 supports only version 14.1 of the MITRE ATT&CK technique list.
2. In the KUMA web interface, go to the **Settings** → **Other** → **General** section.
3. In the **MITRE Techniques List Settings**, click **Import from file**.

   This opens the file selection window.
4. Select the downloaded MITRE ATT&CK technique list and click **Open**.

   This closes the file selection window.

The list of MITRE ATT&CK techniques is imported into KUMA. The list of imported techniques and the version of the MITRE ATT&CK technique list will be displayed by clicking **View list**.

Mapping MITRE ATT&CK techniques to correlation rules

To map MITRE ATT&CK techniques to correlation rules:

1. In the KUMA web interface, go to the **Resources** → **Correlation rules** section.
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 web interface, in the **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 item, 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 techniques to MITRE ATT&CK Navigator:

1. In the KUMA web interface, go to the **Resources** → **Correlation rules** section.
2. Click the ⋯ button in the upper-right corner.
3. In the drop-down list, click **Export to MITRE ATT&CK Navigator**.
4. This opens a window; in that window, 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.

Sigma rules converter

The Sigma rule converter converts correlation rules into a format that KUMA can parse and facilitates importing simple rules to be refined by an analyst.

The Sigma Rule Converter is supported on the following operating systems:

- Ubuntu 2024
- Windows 10, 11

- RED OS 7, 8
- Oracle Linux 8.3, 9.3

To use the Sigma rule converter to convert correlation rules, follow these steps:

1. Install the Sigma rule converter. The following installation options are available:

   - **Installing the Sigma rule converter using docker and bash**

     This installation method is suitable for local installation of the converter web interface.

     Installation requires the latest versions of bash and docker. For detailed information about deploying bash and docker, refer to the official documentation.

     Astra Linux 1.7 is not supported.

     To install a converter:

     1. Enter the directory with the extracted archive and execute one of the following commands:

        - To install the converter on host `0.0.0.0:8000`, run the following command:

          ```
          ./run.sh
          ```

        - To install the converter on host `0.0.0.0:8888`, run the following command:

          ```
          ./run.sh <port>
          ```

          where:
          `<port>` is the port (the default port is `8000`).

        - To install the converter on host `127.0.0.1:9999`, run the following command:

          ```
          ./run.sh <IP address>:<port>
          ```

          where:

          - `<IP address>`

            in the range of
            is the IP address of the listening interface.

            `0.0.0.0`
            . Specify

            `127.0.0.1`
            to run only on the local interface.

          - `<port>`

            is the port.

     The converter is installed.

- **Installing the Sigma rule converter in an environment that allows running docker images**

  This installation method uses the provided container and lets you create a systemd service and deploy the web interface using Kubernetes.

  Any container tool will do, such as docker, podman, or cri-o. Installation involves downloading the container image and then running the image with a port specified.

  To install the converter using podman:

  1. Download the container image:

     ```
     podman load --input "kuma-sigma-converter-x.x.x.tar
     ```

  2. Run the image in one of the following ways:

     - By running the following command:

       ```
       podman run --name=kuma-converter -p "8000:8000" -e "PORT=8000" --rm -it kuma-sigma-converter:x.x.x
       ```

       Wait for the command to finish or to become idle. Do not press **Ctrl**+**C** before the installation process has finished or you get an explicit prompt to abort the installation

     - Run in the background:

       ```
       podman run --name=kuma-converter -p "8000:8000" -e "PORT=8000" --rm kuma-sigma-converter:x.x.x
       ```

  The converter is installed.

- **Installing the Sigma rule converter using Python 3.10 or later**

  This installation method is the most flexible and suitable for experienced users because it lets you adapt the converter to your specific needs. Unlike other methods, this method requires an internet connection to download Python packages.

  To install a converter:

  1. Make sure you have Python 3.10 or later installed.

  2. Install poetry for Python dependency management. For Ubuntu or Debian, you can use the package manager:

     ```
     sudo apt-get update
     sudo apt install pipx
     pipx install poetry
     ```

  3. Use poetry to configure and run the web interface of the Sigma rule converter:

     ```
     cd sigconverter.io
     poetry install
     poetry run ./run.py
     ```

4. In the browser, navigate to the host where the web interface of the Sigma rule converter is installed. For a local installation, the address is `http://127.0.0.1:8000` or `http://0.0.0.0:8000`.

5. If necessary, specify the `HOST` and `PORT` environment variables to configure the listening interface and set the port number.

The converter is installed

.

The Sigma rule converter is a plug-in for the pySigma framework, which means pySigma-backend-kuma can be used with other standard Sigma tools, not only a customized version of sigconverter.io. For example:

- sigma-cli is a converter with a command line interface.
- sigconverter.io is the original generic web interface.

README files for the libraries provided as source code:

- `src/pySigma/README.md` for the industry-standard sigma conversion framework and library.
- `src/pySigma-backend-kuma/README.md` for the plug-in for pySigma that implements KUMA filter and query conversion.
- `src/sigconverter.io/README.md` for the web interface.

After installation, log in to the web interface of the converter.

2. Modify the correlation rule as necessary in the Sigma rule converter and copy or import the modified rule into KUMA.

3. Restart the correlator to apply the changes.

Your edited correlation is applied in the correlator.

Preparing a Sigma rule

To convert a Sigma rule to a KUMA rule:

1. Before modifying the rule, make sure that the following requirements are satisfied:

   - The predefined normalizers use event enrichment `to lower case` for fields containing user names, host names, and process names, therefore we recommend converting the relevant fields to lower case in the source Sigma rule.
   - We do not recommend writing comments or additional information in the `contains` conditions and similar. If you need to leave a comment, we recommend using the # character, such comments do not affect the condition.
   - The rule must begin with the word `title` without hyphens before it.

2. Copy the correlation rule that you want to edit from the KUMA web interface, convert it to YML format, and paste it into the **rule.yml** tab of the converter. Select the settings in the drop-down lists as necessary. Fix any errors if necessary.

3. In the Sigma rule converter, in the **Backend** field, select **kuma**, in the **Format** field, select **default**, and leave the **Pipeline** field empty.

   The following values are available:

   - Backend: kuma, Format: default - ready-to-run SQL query without formatting.
   - Backend: kuma, format: SQL_formatted - ready-to-run SQL query with formatting.
   - Backend: kuma_filter, Format: default - ready-to-use selector for the correlation rule.
   - Backend: kuma_filter, Format: correlation -ready-to-import correlation rule in JSON format. To perform the import, you can use the REST API method /api/v2.1/resources/correlationRule/create and specify the tenantID and the name of the tenant for the import.

4. Go to the tab of the **custom_pipeline.yml** converter and copy the "**Example dictionary adapted for pySigma**" to that tab.

   - Windows process_creation mapping

     product: windows

     category: process_creation

     DeviceEventClassID = 4688

   - Windows sysmon EVENT=11 mapping

     product: windows

     service: Sysmon

     EventID = 11

   - Windows security

     product: windows

     service: security

     EventID = 4624|4662|4104|4698

```
name: KUMA windows
priority: 30
transformations:
### Windows process_creation mapping

  - id: MAPPING_logsource_windows_process_creation
    type: field_name_mapping
    mapping:
      ParentImage: SourceProcessName
      Image: DestinationProcessName
      CommandLine: DeviceCustomString4
    rule_conditions:
      - type: logsource
        product: windows
        category: process_creation

  - id: DEFAULT_FOR_logsource_windows_process_creation
    type: add_condition
    conditions:
      DeviceVendor: Microsoft
      DeviceProduct: Windows
      DeviceEventClassID: '4688'
```

```
    rule_conditions:
       - type: logsource
         product: windows
         category: process_creation
### Windows sysmon EventID=11 mapping
  - id: DEFAULT_FOR_logsource_windows_sysmon
    type: add_condition
    conditions:
      DeviceVendor: Microsoft
      DeviceProduct: Sysmon
    rule_conditions:
       - type: logsource
         product: windows
         service: Sysmon

  - id: MAPPING_logsource_windows_sysmon
    type: field_name_mapping
    mapping:
      EventID: DeviceEventClassID
      Image: OldFilePath
      CommandLine: DeviceCustomString4
      TargetFilename: FilePath
    rule_conditions:
      - type: logsource
        product: windows
        service: Sysmon

### Windows security mapping
  - id: DEFAULT_FOR_logsource_windows_security
    type: add_condition
    conditions:
      DeviceVendor: Microsoft
      DeviceProduct: Windows
    rule_conditions:
       - type: logsource
         product: windows
         service: security
  - id: DEFAULT_MAPPING_FOR_logsource_windows_security
    type: field_name_mapping
    mapping:
      Computer: DestinationHostName
      LogonID: FlexString1
    rule_conditions:
       - type: logsource
         product: windows
         service: security

  - id: 4624_CONDITIONAL_MAPPING_logsource_windows_security
    type: field_name_mapping
    mapping:
      EventID: DeviceEventClassID
      LogonType: DeviceCustomNumber1
      SourceIP: SourceAddress
    rule_conditions:
      - type: contains_detection_item
        field: "EventID"
        value: 4624
```

```
- id: 4662_CONDITIONAL_MAPPING_logsource_windows_security
  type: field_name_mapping
  mapping:
    EventID: DeviceEventClassID
    Properties: FilePermission
  rule_conditions:
    - type: contains_detection_item
      field: "EventID"
      value: 4662

- id: 4104_CONDITIONAL_MAPPING_logsource_windows_security
  type: field_name_mapping
  mapping:
    EventID: DeviceEventClassID
    ScriptBlockText: Message
  rule_conditions:
    - type: contains_detection_item
      field: "EventID"
      value: 4104

- id: 4698_CONDITIONAL_MAPPING_logsource_windows_security
  type: field_name_mapping
  mapping:
    EventID: DeviceEventClassID
    TaskName: SourceProcessName
  rule_conditions:
    - type: contains_detection_item
      field: "EventID"
      value: '4698'
```

The converted rule is displayed on the **query** tab. The result of the conversion is a `WHERE` clause for an SQL query.

You can copy the end result to KUMA if it is a query or a filter. If you want to use a rule in KUMA, you can import rules into KUMA one by one.

Examples of supported Sigma rule conversions

Examples of supported Sigma rule conversions

Sigma rule	Mapping to KUMA objects	
title: Generic-DCSync Execution id: 9107f781-c984-436a-8093-1aaa64c70c49 description: Detects access to DC with "DS-Replication-Get-Changes" or "DS-Replication-Get-Changes-All" rights author: Kaspersky status: stable modified: 2023-08-02 tags: - attack.credential_access - attack.t1003.006 logsource: product: windows service: security detection: selection1: EventID: 4662 Properties|contains: - 'DS-Replication-Get-Changes' - '1131f6aa-9c07-11d1-f79f-00c04fc2dcd2' - '1131f6ad-9c07-11d1-f79f-00c04fc2dcd2' selection2: EventID: 4624 LogonType: 3 filter: EventID: 4624 LogonType: 3 SourceIP: - '<ip address of DC servers>' timeframe: 1m condition: selection1 or (selection2 and not filter) | count(EventID) by Computer,LogonID = 2 fields: - SourceIP - Computer - LogonID - Username - Properties falsepositives: - Unknown level: high	"Microsoft products" normalizer tags → description logsource DeviceVendor = Microsoft DeviceProduct = Windows selection1: DeviceEventClassID=4662 FilePermission contains '' OR FilePermission contains '' OR FilePermission contains '' selection2 DeviceEventClassID=4624 DeviceCustomNumber1 = 3 filter DeviceEventClassID=4624 DeviceCustomNumber1 = 3 SourceAddress = <IP> condition: DeviceVendor = Microsoft AND DeviceProduct = Windows AND ((DeviceEventClassID=4662 FilePermission contains '' OR FilePermission contains '' OR FilePermission contains '') OR ((DeviceEventClassID=4624 AND DeviceCustomNumber1 = 3) NOT (DeviceEventClassID=4624 AND DeviceCustomNumber1 = 3 AND SourceAddress = <IP>))) everything inside Count() Pass EventID to Unique fields, everything after 'by' Computer → DestinationHostName LogonID → FlexString1 pass to Aggregation fields, everything after '=' or '>' pass 2 to threshold. timeframe pass to time window	Sigma correlation remove the pipe o converted into a c SQL query for Thr

Sigma rule	Mapping to KUMA objects	
title: Lsass Dump via LOLBin id: 2fe9cd33-d7f1-4d52-ab11-e40cb359ad02 description: detects lsass dump via lolbins such as procdump.exe, dotnet-dump.exe, dumpminitool.exe references: - https://twitter.com/bohops/status/1635288066909966338 - https://twitter.com/mrd0x/status/1511415432888131586 modified: 2023-07-18 author: Kaspersky status: stable tags: - attack.credential_access - attack.t1003.001 logsource: product: windows category: process_creation detection: selection_procdump: Image|endswith: - '\procdump.exe' - '\procdump64.exe' CommandLine|contains: 'lsass' selection_dotnet: Image|endswith: '\dotnet-dump.exe' CommandLine: ' collect ' selection_dumpminitool: Image|endswith: '\dumpminitool.exe' condition: 1 of selection* falsepositives: - Unknown level: high	Converted in the same way as the rule in row 3 of the table. The difference is: the `condition` uses a `1 of selection` construct, where `selection` denotes one of the named groups of conditions.	DeviceVendor=' DeviceProduct= DeviceEventCla (((Desti '\\procdump.ex '\\procdump64. 'lsass') OR (Destinati dump.exe' AND De) OR Destinati '\\dumpminitoo)
title: Generic-Clearing Windows Event Logs via Command Line id: a903492e-da40-4ab4-a92d-306a4799d973 description: Detects Clear Windows Event Logs via Command Line author: Kaspersky status: stable modified: 2023-07-18 tags: - attack.defense_evasion - attack.t1070.001 logsource: product: windows category: process_creation detection: selection1: Image|endswith: '\wevtutil.exe' CommandLine|contains: - ' cl ' - 'clear' selection2: Image|endswith: - '\powershell.exe' - '\powershell_ise.exe' - '\pwsh.exe' CommandLine|contains: - 'Clear-EventLog' - 'Remove-EventLog' selection3: Image|endswith: '\wmic.exe' CommandLine|contains: - 'ClearEventLog' condition: 1 of them falsepositives: - Legitimate System Administrator actions level: high	"Microsoft products" normalizer tags → description logsource DeviceVendor = Microsoft DeviceProduct = Windows DeviceEventClassID=4688 selection1: DestinationProcessName endswith '' DeviceCustomString4 contains "" OR DeviceCustomString4 contains "" Selection2 DestinationProcessName endswith '' DeviceCustomString4 contains "" OR DeviceCustomString4 contains "" selection3 DestinationProcessName endswith '' DeviceCustomString4 contains "" OR DeviceCustomString4 contains "" condition DeviceVendor = Microsoft AND DeviceProduct = Windows AND DeviceEventClassID=4688 AND ((DestinationProcessName endswith '' AND (DeviceCustomString4 contains "" OR DeviceCustomString4 contains "")) OR (DestinationProcessName endswith '' AND (DeviceCustomString4 contains "" OR DeviceCustomString4 contains "")) OR (DestinationProcessName endswith '' AND (DeviceCustomString4 contains "" OR DeviceCustomString4 contains ""))) identical fields – specify the fields from the condition, except Name, deviceProduct, deviceVendor	DeviceVendor=' DeviceProduct= DeviceEventCla ((Destinati '\\wevtutil.ex (Devic DeviceCustomSt) OR ((Destin '\\powershell. Des '\\powershell\ '\\pwsh.exe') A (Device EventLog' OR D 'Remove-EventL) OR (Destinatio AND DeviceC 'ClearEventLog)

Sigma rule	Mapping to KUMA objects	
title: 0021-0002.1204.003 Execution - PowerExchangeDropper description: Detect the dropping of PowerExchange components. status: experimental logsource: product: windows service: Sysmon detection: selection1: EventID: 11 selection2: TargetFilename|contains: - 'c:\users\public\microsoftedge\autosave.exe' - 'c:\users\public\microsoftedge\wsdl.ps1' selection3: Image|endswith: "brochure.exe" condition: selection1 and selection2 and selection3 falsepositives: - Unknown level: high	Based on: Normalizer 11 of Sysmon event. title → name author → description description→ description status → description logsource DeviceVendor = Microsoft DeviceProduct = Sysmon selection1 DeviceEventClassID=11 selection2 FilePath contains 'c:\users\public\microsoftedge\autosave.exe' OR FilePath contains '' if the file path contains * format in regexp selection3 OldFilePath endswith "" condition: DeviceVendor = Microsoft AND DeviceProduct = Sysmon AND (DeviceEventClassID=11) AND (FilePath contains 'c:\users\public\microsoftedge\autosave.exe' OR FilePath contains '') AND (OldFilePath endswith "") identical fields – specify the fields from the condition, except Name, deviceProduct, deviceVendor level →severity falsepositive → Description	DeviceVendor=' DeviceProduct= DeviceEventCla (FilePath con 'c:\\users\\pu OR FilePath c 'c:\\users\\pu) AND OldFilePath en
title: Domain joined host discovery description: Detect an attacker enumerating domain-joined machines using PowerShell. status: stable logsource: product: windows service: security detection: selection: EventID: 4104 ScriptBlockText|contains|all: - 'System.DirectoryServices.DirectorySearcher' - 'Properties.operatingsystem' condition: selection falsepositives: - "Administrative scripts or tools like Vulnerability Scanners" level: medium	ScriptBlockText → Message	

Sigma rule	Mapping to KUMA objects	
title: 0022-0002.1053.005 Execution - PowerExchange Task description: Detect the creation of a scheduled task by PowerExchange. status: experimental logsource: product: windows service: security detection: selection: EventID: - '4698' TaskName|contains: - 'MicrosoftEdgeUpdateService' condition: selection falsepositives: - Unknown level: medium	TaskName → SourceProcessName	

Sigma rule	Mapping to KUMA objects	
title: Created Windows Shell from Critical Windows Process description: Anomaly behavior critical windows process status: stable modified: 2023-07-18 tags: - attack.defense_evasion - attack.t1036 logsource: product: windows category: process_creation detection: selection: ParentImage|endswith: - '\searchindexer.exe' - '\lsaiso.exe' - '\lsm.exe' - '\spoolsv.exe' - '\wininit.exe' - '\smss.exe' - '\csrss.exe' - '\lsass.exe' - '\services.exe' - '\winlogon.exe' Image|endswith: - '\powershell_ise.exe' - '\cmstp.exe' - '\appvlp.exe' - '\mftrace.exe' - '\scriptrunner.exe' - '\forfiles.exe' - '\msiexec.exe' - '\rundll32.exe' - '\mshta.exe' - '\hh.exe' - '\wmic.exe' - '\regsvr32.exe' - '\scrcons.exe' - '\bash.exe' - '\sh.exe' - '\cscript.exe' - '\wscript.exe' - '\powershell.exe' - '\cmd.exe' condition: selection falsepositives: - Unknown level: high	ParentImage → SourceProcessName	

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. Events that satisfy the filter conditions are sent to KUMA for further processing.

You can use filters in the following KUMA services and features:

- Collector.
- Correlator.
- Storage.
- 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 in which they were created. For resources in input fields except the **Description** field, you can enable the display of control characters. Available filter settings are listed in the table below.

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting. Inline filters are created in other resources or services and do not have names.
Tenant	The name of the tenant that owns the resource. Required setting.
Tags	Tags for resource search. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

You can create filter conditions and filter groups, or add existing filters to a filter.

To create filtering criteria, you can use *builder mode* or *source code mode*. 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. The builder mode is used by default.

You can freely switch between modes when creating filtering criteria. To switch to source code mode, select the **Code** tab. 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 to display the filter code.

Creating filtering criteria in builder mode

To create filtering criteria in builder mode, you need to select one of the following operators from the drop-down list:

- **AND**: The filter selects events that match all of the specified conditions.
- **OR**: The filter selects events that match one of the specified conditions.
- **NOT**: The filter selects events that match none of the specified conditions.

You can add filtering criteria in one of the following ways:

- To add a condition, click the **+ Add condition** button.
- To add a group of conditions, click the **+ Add group** button. When adding groups of conditions, you can also select the **AND**, **OR**, and **NOT** operators. In turn, you can add conditions and condition groups to a condition group.

You can add multiple filtering criteria, reorder the filtering criteria, or remove filtering criteria. To reorder filtering criteria, use the reorder ⣿ icons. To remove a filtering criterion, click the delete ✕ icon next to it.

Available condition settings are listed in the table below.

Setting	Description
<Condition type>	Condition type. The default is **If**. You can click the default value and select **If not** from the displayed drop-down list. Required setting.
<Left operand> and **<Right operand>**	Values to be processed by the operator. The available types of values of the right operand 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. Required settings.

Setting	Description
<Operator>	Condition operator. When selecting a condition operator in the drop-down list, you can select the **do not match case** check box if you want the operator to ignore the case of values. This check box is ignored if the **inSubnet**, **inActiveList**, **inCategory**, **InActiveDirectoryGroup, hasBit**, and **inDictionary** operators are selected. By default, this check box is cleared. **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**—presence in the left operand of the list items specified in the right operand. You can change or delete the specified operator. To change the operator, click it and specify a new operator. To delete the operator, click it, then press **Backspace**.

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 of the `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.

Creating filtering criteria in source code mode

The source code mode allows you to quickly edit conditions, select and copy blocks of code. In the right part 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 separate 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 "Read shared resources" role. To view the list of resources for the selected operand inside the expression, press **`Ctrl+Space`**. This displays a list of resources.

The filters listed in the table below are included in the KUMA 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. The active list must be linked to a correlation rule, and the correlation rule must be linked to a correlator.

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.

> If the correlator is disabled while processing active list data, the "Internal communication failed" error is displayed in the interface, indicating that the connection to the data source, namely the correlator, has been lost.

During the correlation process, when entries are deleted from active lists after their lifetime expires, 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 processed and 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 correlator active lists

To view the table of correlator active lists:

1. In the KUMA web interface, select **Resources** → **Work with services** → **Active services**.
2. Select the check box next to the correlator for which you want to view the active list.
3. In the toolbar or in the menu (⋮), click **Go to active lists**.

   The **Correlator active lists** table is displayed.

Description of the Correlator active lists table

Column name	Description
Name	Active list name.
Records	The number of records in the active list.
Size on disk	Disk space occupied by the active list, KB.
Directory	Path to the active list on the KUMA Core server.

Adding active list

To add active list:

1. In the KUMA web interface, select the tenants in which you want to manage active lists.
2. In the KUMA web interface, select the **Resources** → **Resources** → **Active lists** section.
3. In the toolbar or in the menu (⋮), click **Add active list**.
4. 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 in seconds for which the entry added to the active list must remain 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 (approximately 100 years).
   d. If necessary, in the **Tags** field, specify the tags to help find the resource.
   e. If necessary, enter additional information in the **Description** field.

      You can use up to 4,000 Unicode characters.

5. To save the active list, do one of the following:

   - Click the **Save** button.
   - Click the **Save with comment** button.

     If necessary, add a comment in the **Leave a comment** window that is displayed. To view comments for an active list, in the table of active lists, select the check box next to an active list and in the toolbar or in the menu (⋮) click **View versions history**. This opens the **Version history** window with a table of available versions of the active list, in which you can view comments.

The active list is added.

Viewing the settings of an active list

To view the settings of an active list:

1. In the KUMA web interface, select the tenants in which you want to manage active lists.
2. In the KUMA web interface, select **Resources** → **Resources** → **Active lists**.

   A table of available active lists is displayed.
3. In the **Name** column, select the active list whose settings you want to view.

The settings of the active list are displayed. For a description of active list settings, see the following table.

Settings of an active list

Setting name	Description
ID	Active list ID.
Name	Unique name of the active list
Tenant	Name of the tenant that owns the active list
TTL	Retention period of a record added to the active list. This value is specified in seconds.
Tags	Search tags of the active list
Description	Any additional information about the active list
Correlator	Name of the correlator to which the active list is linked. This field is displayed if the active list is linked to a correlator.

If the active list is linked to a correlator, the table of values of the active list is displayed.

Changing the settings of an active list

To change the settings of an active list:

1. In the KUMA web interface, select the tenants in which you want to manage active lists.
2. In the KUMA web interface, select **Resources** → **Resources** → **Active lists**.

   A table of available active lists is displayed.
3. In the **Name** column, select the active list whose settings you want to change.

4. Make the necessary changes to the following parameters:

   - **Name**—unique name of the active list
   - **TTL**—time to live parameter of entries stored in the Active list, in seconds.

     If the value of the field is 0, the record is retained for 36,000 days (approximately 100 years).
   - **Tags**—Search tags of the active list
   - **Description**—any additional information about the active list

   The **ID**, **Tenant**, and **Correlator** fields are not editable.

Duplicating an active list

To copy an active list:

1. In the KUMA web interface, select the tenants in which you want to manage active lists.
2. In the KUMA web interface, select **Resources** → **Resources** → **Active lists**.
3. Select the check box next to the active lists you want to copy.
4. In the toolbar or in the menu (⋮), click **Duplicate**.
5. Specify the name of the list and the tenant to which you want to copy the active list. By default, the active list that you are duplicating gets a name in the <Active_list_name>-copy format , and it is duplicated in the tenant that you are currently managing.
6. To save the active list, do one of the following:

   - Click the **Save** button.
   - Click the **Save with comment** button.

     If necessary, add a comment in the **Leave a comment** window that is displayed. To view comments for an active list, in the table of active lists, select the check box next to an active list and in the toolbar or in the menu (⋮) click **View versions history**. This opens the **Version history** window with a table of available versions of the active list, in which you can view comments.

The active list is copied.

Deleting active lists

To delete active lists:

1. In the KUMA web interface, select the tenants in which you want to manage active lists.
2. In the KUMA web interface, select **Resources** → **Resources** → **Active lists**.
3. Select the check boxes next to the active lists you want to delete.

If you want to delete all active lists, select the check box next to the **Name** column and in the context menu, select one of the following options:

- **Select all**
- **Select everything on the page**

> At least one check box must be selected.

4. Click the **Delete** button.
5. In the confirmation window, click **Delete**.

   If there are any links to the selected active lists from other resources and services, such lists cannot be deleted. In this case, you must first unlink the active lists from the correlation rules and correlators.

The active lists are deleted.

Viewing records in the active list

To view the records in the active list:

1. In the KUMA web interface, select the tenants in which you want to manage active lists and do one of the following:

   - Go to the **Resources** → **Resources** → **Active lists** section.

     A table of available active lists is displayed.

   - Go to the **Resources** → **Work with services** → **Active services** section.

     Select the check box next to the correlator for which you want to view the active list.

     In the toolbar or in the menu (⋮), click **Go to active lists**.

     The **Correlator active lists** table is displayed.

2. In the **Name** column, select the desired active list.

A table of records for the selected active list is opened. If the active list is linked to a correlator, the table of values of the active list is displayed.

Description of the table of active list records

Setting name	Description
Key	Name of the active list key. Multiple values can be specified using the | character.
Value	Value of the active list key. You can specify multiple key values separated by commas in JSON format.
Actions	Actions that can be performed with active list records. You can edit or delete records.
Record repetitions	Total number of times the record was mentioned in events and identical records were downloaded when importing active lists to KUMA.
Expires	Date and time when the record will 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 (approximately 100 years).
Updated	The time when the active list was last updated.

The page can display up to 64 active list records. If necessary, you can scroll the page to view more records.

Searching for records in the active list

To find a record in the active list:

1. In the KUMA web interface, select the tenants in which you want to manage active lists and do one of the following:

   - Go to the **Resources** → **Resources** → **Active lists** section.

     A table of available active lists is displayed.

   - Go to the **Resources** → **Work with services** → **Active services** section.

     Select the check box next to the correlator for which you want to view the active list.

     In the toolbar or in the menu (⋮), click **Go to active lists**.

     The **Correlator active lists** table is displayed.

2. In the **Name** column, select the desired active list.

   A table of records for the selected active list is opened.

3. In the **Search** field, enter the record key name or several characters from the name. You can also search by key value.

The table of records of the active list displays only the records with the value containing the entered characters.

Adding a record to an active list

To add a record to the active list:

1. In the KUMA web interface, select the tenants in which you want to manage active lists and select one of the following options:

   - Go to the **Resources** → **Resources** → **Active lists** section.

     A table of available active lists is displayed.

   - Go to the **Resources** → **Work with services** → **Active services** section.

     Select the check box next to the correlator for which you want to view the active list.

     In the toolbar or in the menu (⋮), click **Go to active lists**.

     The **Correlator active lists** table is displayed.

2. In the **Name** column, select the desired active list.

   A table of records for the selected active list is opened.

3. Click **Add**.

4. 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.

   b. In the **Value** field, specify a value in JSON format. You can specify multiple values.

5. Click the **Save** button.

The record is added. After saving, the records in the active list are sorted in alphabet order.

The page can display up to 64 active list records. If necessary, you can scroll the page to view more items.

Duplicating records in the active list

To duplicate a record in the active list:

1. In the KUMA web interface, select the tenants in which you want to manage active lists and do one of the following:

   - Go to the **Resources** → **Resources** → **Active lists** section.

     A table of available active lists is displayed.

   - Go to the **Resources** → **Work with services** → **Active services** section.

     Select the check box next to the correlator for which you want to view the active list.

     In the toolbar or in the menu (⋮), click **Go to active lists**.

     The **Correlator active lists** table is displayed.

2. In the **Name** column, select the desired active list.

   A table of records for the selected active list is opened.

3. Select the check box next to the record that you want to copy.

4. Click **Duplicate**.

5. Specify the key and, if necessary, change the values of the key.

6. Click the **Save** button.

The record is copied. After saving, the records in the active list are sorted in alphabet order.

The page can display up to 64 active list records. If necessary, you can scroll the page to view more items.

Changing a record in the active list

To edit a record in the active list:

1. In the KUMA web interface, select the tenants in which you want to manage active lists and do one of the following:

   - Go to the **Resources** → **Resources** → **Active lists** section.

     A table of available active lists is displayed.

   - Go to the **Resources** → **Work with services** → **Active services** section.

     Select the check box next to the correlator for which you want to view the active list.

     In the toolbar or in the menu (⋮), click **Go to active lists**.

     The **Correlator active lists** table is displayed.

2. In the **Name** column, select the desired active list.

   A table of records for the selected active list is opened.

3. Click the pencil icon in the **Actions** column.

4. Edit key values in JSON format.

5. If you want to confirm your changes, click the check mark in the **Actions** column. If you want to discard your changes, click the X mark in the **Actions** column.

6. Click **Save**.

The record is overwritten. After saving, the records in the active list are sorted in alphabet order.

The page can display up to 64 active list records. If necessary, you can scroll the page to view more items.

Deleting records from the active list

To delete records from the active list:

1. In the KUMA web interface, select the tenants in which you want to manage active lists and do one of the following:

   - Go to the **Resources** → **Resources** → **Active lists** section.

     A table of available active lists is displayed.

   - Go to the **Resources** → **Work with services** → **Active services** section.

     Select the check box next to the correlator for which you want to view the active list.

     In the toolbar or in the menu (⋮), click **Go to active lists**.

     The **Correlator active lists** table is displayed.

2. In the **Name** column, select the desired active list.

   A table of records for the selected active list is opened.

3. Select the check boxes next to the records that 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.

4. Click the **Delete** button.

5. Click **OK**.

   The records will be deleted.

The page can display up to 64 active list records. If necessary, you can scroll the page to view more items.

Import entries to an active list

To import records into an active list:

1. In the KUMA web interface, select the tenants in which you want to manage active lists and do one of the following:

   - Go to the **Resources** → **Resources** → **Active lists** section.

     A table of available active lists is displayed. Click the name of an active list to open the **Edit active list** window, and in that window, in the **Content** section, select the check boxes next to the records that you want to import.

   - Go to the **Resources** → **Work with services** → **Active services** section.

     Select the check box next to the correlator for which you want to view the active list.

     In the toolbar or in the menu (⋮), click **Go to active lists**.

     In the **Correlator active lists** table that is displayed, select the check box next to the active sheet that you need.

2. In the toolbar, click **Import**.

   The active list import window opens.

3. Click the **Add** button to open a file selection window and select the file that you want to import.

4. In the **Format** drop-down list select the format of the file:

   - **csv**
   - **tsv**
   - **internal**

5. Under **Key field**, enter the name of the column containing the keys of records of the active list specified in the file. This field is not available for the **internal** format.

6. 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 records from the active list

To export records from the active list:

1. In the KUMA web interface, select the tenants in which you want to manage active lists and do one of the following:

   - Go to the **Resources** → **Resources** → **Active lists** section.

     A table of available active lists is displayed. Click the name of an active list to open the **Edit active list** window, and in that window, in the **Content** section, select the check boxes next to the records that you want to export.

   - Go to the **Resources** → **Work with services** → **Active services** section.

     Select the check box next to the correlator for which you want to view the active list.

     In the toolbar or in the ⋮ menu, click **Go to active lists**.

     In the **Correlator active lists** table that is displayed, select the check box next to the active sheet that you need.

2. In the toolbar, click **Export**.

The active list is downloaded in JSON format to the default folder or to the folder specified in your browser 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 KUMA 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 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.

Proxies

Proxy servers are used to store proxy server configuration settings, for example, in destinations. The http type is supported. Available proxy server settings are listed in the table below.

Available proxy server settings

Setting	Description
Name	Unique name of the proxy server. Maximum length of the name: 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Secret separately	Viewing information about the connection. If this check box is selected, the following settings are displayed in the window: • **URL** is the connection URL. • **Secret** is the secret of the 'credentials' type. This lets you 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. This check box is cleared by default.
Use URL from the secret	The secret resource that stores URLs of proxy servers. You can create or edit a secret. To create a secret, click the plus (+) icon. To edit a secret, click the pencil ( ) icon. We do not recommend using this option. Instead, manually specify the URL and the secret separately by selecting the **Secret separately** check box.
Do not use for domains	One or more domains to which direct access is required.
Description	Description of the proxy server. Maximum length of the description: 4000 Unicode characters.

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 dictionary settings are listed in the table below.

Available dictionary settings

Setting	Description
Name	Unique name for this resource type. Maximum length of the name: 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Description	Description of the resource. Maximum length of the description: 4000 Unicode characters.
Type	Dictionary type. The selected dictionary type determines the format of the data that the dictionary can contain: • You can add key-value pairs to the **Dictionary** type. We do not recommend adding more than 50,000 entries to dictionaries of this type using the KUMA web interface. 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 dictionaries of this type using the REST API. When adding dictionaries using the API, there is no limit on the number of entries that can be added. Required setting.
Values	Table with dictionary data. • For the **Dictionary** type, this block displays a list of **Key**—**Value** pairs. You can add and remove rows from the table. To add a row to the table, click +. To remove a row from the table, hover over the table row until the × button appears and click the button. In the **Key** field, you must specify a unique key. Maximum length of the key: 128 Unicode characters. The first character cannot be $. In the **Value** field, you must specify a value. Maximum length of the value: 255 Unicode characters. The first character cannot be $. You may add one or more **Key** – **Value** pairs. • For the **Table** type, this block displays a table containing data. You can add and remove rows and columns from the table. To add a row or column to the table, click +. To remove a row or column from the table, hover over the row or the heading of the column until the × button appears and click the button. You can edit the headings of table columns. If the dictionary contains more than 5,000 entries, they are not displayed in the KUMA web interface. 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 using 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 KUMA 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, KATA, 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, KATA, and Active Directory tasks in accordance with response rules is available when integrated with the relevant programs.

You can configure response rules under **Resources** → **Response**, 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

General 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.
Type	Required setting, available if KUMA is integrated with Kaspersky Security Center. Response rule type, **ksctasks**.
Kaspersky Security Center 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
Handlers	The number of handlers that the service can run simultaneously to process response rules in parallel. By default, the number of handlers 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** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

Setting	Description
	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.

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 solution 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

General 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.
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}}"`
Handlers	The number of handlers that the service can run simultaneously to process response rules in parallel. By default, the number of handlers 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** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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**.

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

General 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.
Type	Required setting. Response rule type, **Response via KICS/KATA**.
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**.
Handlers	The number of handlers that the service can run simultaneously to process response rules in parallel. By default, the number of handlers 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** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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.

Setting	Description
	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 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

General tab

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 the **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 General Administrator role, the **Run program** task will be displayed in the **Task manager** section of the solution web interface. In the **Created by** column of the task table, **Scheduled task** is displayed for this 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 application 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.
Handlers	The number of handlers that the service can run simultaneously to process response rules in parallel. By default, the number of handlers 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** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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**.

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

General 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.
Type	Required setting. Response rule type, **Response via Active Directory**.
Source of the user account ID	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
Group DN	The DistinguishedName of the domain group in fields for each role. The users of this domain group must be able to authenticate with their domain user accounts. Example of entering a group: OU=KUMA users,OU=users,DC=example,DC=domain
Handlers	The number of handlers that the service can run simultaneously to process response rules in parallel. By default, the number of handlers is the same as the number of virtual processors on the server where the service is installed.

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** 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, fields of additional parameters for identifying the value to be passed to the filter may be displayed. 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**.

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 with EDR actions

Starting with KUMA 3.4.1 and Kaspersky Endpoint Security 12.9 for Windows, EDR actions are supported when responding to threats. When correlation events are received, KUMA performs EDR actions. Performing EDR actions requires first setting up response rules. In the properties of response rules, you need to specify the arguments of scripts that you can download by following this link:

https://box.kaspersky.com/d/579c0271a38e440b9144/

When a response rule is triggered, Kaspersky Endpoint Security performs EDR actions and creates the following tasks in Kaspersky Security Center:

- onDemandScan.py performs a Critical Areas Scan.
- ioCScan.py searches for indicators of compromise.
- quarantineFile.py quarantines a file.
- deleteFile.py deletes a file.
- getFile.py gets a file.
- isolateHost.py isolates the computer from the network.
- preventExecution.py prevents execution of objects.

  To perform this task, it is important to make sure that in Kaspersky Security Center, in the **Assets (Devices)** → **Policies & profiles** → <Kaspersky Endpoint Security policy name> → **Application settings** → **Detection and Response** → **Endpoint Detection and Response**, the **Execution Prevention DISABLED** toggle switch is turned off.

- startProcess.py starts a process.
- terminateProcess.py terminates a process

When setting up response rules with EDR actions, we recommend taking into account the load on the computer when running tasks. If the response rules cause scripts to create too many tasks, the computer's performance may be degraded. If there are too many requests, the requests are rotated regardless of whether a particular request has been completed. Kaspersky Endpoint Security allows creating no more than 100 tasks. When this limit is reached, Kaspersky Endpoint Security rotates tasks in Kaspersky Security Center. The lifespan of a task is 30 days.

> Tasks created when the response rules from KUMA are executed are displayed in Kaspersky Endpoint Security of the host on which the response rule is triggered. These tasks are not displayed in Kaspersky Security Center.

You can use reports to view information about the execution of EDR actions in the Administration Console of Kaspersky Security Center. Kaspersky Endpoint Security generates events with descriptions in the '[Response] [kuma] $<script name> - $<date>' format. A description in this format allows creating event selections for EDR actions.

Configuring response rules with EDR actions

Prerequisites

Before configuring response rules with EDR actions, make sure that the following conditions are satisfied:

1. Integration with Kaspersky Security Center is configured.
2. A separate user has been created in Kaspersky Security Center that has the rights to run all Kaspersky Endpoint Security tasks and tasks that you plan to run using scripts.
3. Integration with Kaspersky Endpoint Security for Windows is configured. When installing Kaspersky Endpoint Security, you must install the EDR component.
4. The following actions have been performed on the host with the correlator:
   a. Python 3.6 or later and the KIAkOAPI Python package are installed.
   b. The scripts have been downloaded from the https://box.kaspersky.com/d/579c0271a38e440b9144/ link and placed in /opt/kaspersky/kuma/correlator/<Correlator ID>/scripts.
   c. The scripts are connected to Kaspersky Security Center.

      The credentials are not protected, so we recommend using a separate Kaspersky Security Center user created at step 2 for connecting the scripts to Kaspersky Security Center and restricting access to the directory that contains the scripts. To connect the scripts, open the scripts for editing and edit the lines as follows:

```
# KSC connection info
# reads Environment variables by default, change with corresponding values
# KSC host, string, required
kscHost = "kscHost"
# KSC port, required, default is "13299"
kscPort = "13299"
# KSC certificate file path, default is False
kscCert = False
```

```
# KSC user, string, required
kscUser = "kscUser"
# KSC password, string, required
kscPassword = "kscPassword"
```

Configuring the response

To create a response rule in a KUMA correlator:

1. In the KUMA web interface, in the **Resources and services** → **Correlators** section, create a correlator or open an existing correlator for editing.

2. In the correlator creation wizard, at the **Response** step, click **Add response rule** to open the **Response** window, and in that window, fill in the following fields:

   a. In the **Type** drop-down list, specify the **Run script** response rule type.

   b. In the **Timeout** field, specify the number of seconds allotted for the script to finish. If this value is exceeded, the script is terminated. Set this value to **600** (10 minutes).

   c. In the **Script name** field, enter the name of the script, for example, **deleteFile.py**.

   d. In the **Script arguments** field, specify options or event field values that must be passed to the script. You can use single quotes to specify unescaped characters in options. Arguments of each script are described in expandable boxes.

      onDemandScan.py performs a Critical Areas Scan.

      The onDemandScan script automatically creates and runs the *Critical Areas Scan* task when certain KUMA events are detected. By default, Kaspersky Endpoint Security checks kernel memory, running processes, and boot sectors.

      Limitations

      If the Kaspersky Sandbox or KATA Sandbox solution is deployed, after updating the management plug-in to version 12.9, you need to re-establish the background connection between Kaspersky Security Center Web Console and the Administration Server. This is necessary to update user rights to the new EDR action functionality for threat responsein KUMA.

      Arguments

      To configure the *Critical Areas Scan* task, you need to specify the following arguments for the script:

      `--targetHost` is the ID of the computer on which the script must be run. Required argument. The script gets this argument from the event.

      `--recursive true|false` enables or disables the recursive scanning mode. This argument must be specified manually.

      `--path` is a space-delimited list of files and folders for Custom Scan.

You can also select a predefined scan scope. Kaspersky Endpoint Security has the following predefined objects for configuring the scan scope:

- **System memory**.
- **Running processes and Startup Objects**.

  Memory occupied by processes and executable files of applications that run at operating system startup.
- **Disk boot sectors**.

  Boot sectors of hard drives and removable media.
- **System Backup**.

  Contents of the System Volume Information folder.
- **All external devices**.
- **All hard drives**.
- **All network drives**.

To do this, in the onDemandScan script, for the "enabled" setting of the relevant scan scope, set the value to True or False:

```
defaultPreset = [
    {"enabled": False, "path": "%personal%", "recursive": True,
        "type": ScanObjectType.Folder.value},
    {"enabled": False,
     "path": "", "recursive": True, "type": ScanObjectType.Email.value},
    {"enabled": True,
     "path": "", "recursive": True, "type": ScanObjectType.SystemMemory.value},
    {"enabled": True,
     "path": "", "recursive": True, "type": ScanObjectType.StartupObjectsAndRunningProcesses.value},
    {"enabled": True,
     "path": "", "recursive": True, "type": ScanObjectType.DiskBootSectors.value},
    {"enabled": False,
     "path": "", "recursive": True, "type": ScanObjectType.SystemBackupStorage.value},
    {"enabled": False,
     "path": "", "recursive": True, "type": ScanObjectType.AllRemovableDrives.value},
    {"enabled": False,
     "path": "", "recursive": True, "type": ScanObjectType.AllFixedDrives.value},
    {"enabled": False,
     "path": "", "recursive": True, "type": ScanObjectType.AllNetworkDrives.value}
```

Example

```
python3 onDemandScan.py --targetHost 004c7fcd-8940-4d62-8a94-911932dc861b --recurive
false --path 'C:\Program Files (x86)\Example Folder'
```

The onDemandScan script automatically creates and runs the *Critical Areas Scan* task when certain KUMA events are detected. By default, Kaspersky Endpoint Security checks kernel memory, running processes, and boot sectors.

Limitations

If the Kaspersky Sandbox or KATA Sandbox solution is deployed, after updating the management plug-in to version 12.9, you need to re-establish the background connection between Kaspersky Security Center Web Console and the Administration Server. This is necessary to update user rights to the new EDR action functionality for threat responsein KUMA.

Arguments

To configure the *Critical Areas Scan* task, you need to specify the following arguments for the script:

`--targetHost` is the ID of the computer on which the script must be run. Required argument. The script gets this argument from the event.

`--recursive true|false` enables or disables the recursive scanning mode. This argument must be specified manually.

`--path` is a space-delimited list of files and folders for Custom Scan.

You can also select a predefined scan scope. Kaspersky Endpoint Security has the following predefined objects for configuring the scan scope:

- **System memory**.
- **Running processes and Startup Objects**.

  Memory occupied by processes and executable files of applications that run at operating system startup.
- **Disk boot sectors**.

  Boot sectors of hard drives and removable media.
- **System Backup**.

  Contents of the System Volume Information folder.
- **All external devices**.
- **All hard drives**.
- **All network drives**.

To do this, in the onDemandScan script, for the "enabled" setting of the relevant scan scope, set the value to True or False:

```
defaultPreset = [
    {"enabled": False, "path": "%personal%", "recursive": True,
        "type": ScanObjectType.Folder.value},
    {"enabled": False,
     "path": "", "recursive": True, "type": ScanObjectType.Email.value},
    {"enabled": True,
     "path": "", "recursive": True, "type": ScanObjectType.SystemMemory.value},
    {"enabled": True,
     "path": "", "recursive": True, "type": ScanObjectType.StartupObjectsAndRunningProcesses.value},
    {"enabled": True,
     "path": "", "recursive": True, "type": ScanObjectType.DiskBootSectors.value},
    {"enabled": False,
     "path": "", "recursive": True, "type": ScanObjectType.SystemBackupStorage.value},
    {"enabled": False,
     "path": "", "recursive": True, "type": ScanObjectType.AllRemovableDrives.value},
    {"enabled": False,
     "path": "", "recursive": True, "type": ScanObjectType.AllFixedDrives.value},
    {"enabled": False,
     "path": "", "recursive": True, "type": ScanObjectType.AllNetworkDrives.value}
```

Example

```
python3 onDemandScan.py --targetHost 004c7fcd-8940-4d62-8a94-911932dc861b --recurive
false --path 'C:\Program Files (x86)\Example Folder'
```

ioCScan.py searches for indicators of compromise.

The iocScan script automatically creates and runs the *IOC Scan* task when certain KUMA events are detected. The *IOC Scan* task allows detecting indicators of compromise on the computer and performing threat response actions.

Kaspersky Endpoint Security uses IOC files to search for indicators of compromise. *IOC files* contain a set of indicators that are matched against events to generate detections. IOC files must conform to the OpenIOC standard.

Limitations

By default, Kaspersky Endpoint Security performs the IOC scan only in critical areas of the computer, such as the Downloads folder, the Desktop, the temporary files folder of the operating system, and others.

By default, the following event logs are selected: Application Event Log, System Event Log, and Security Event Log.

Arguments

To configure the *IOC Scan* task, you need to specify the following arguments for the script:

`--targetHost` is the ID of the computer on which the script must be run. Required argument. The script gets this argument from the event.

`--iocZip` is the path to the ZIP archive with base64-encoded IOC files that you want to use for the scan. Required argument. This argument must be specified manually.

`--isolateHost true|false` specifies whether the computer must be isolated from the network when an indicator of compromise is detected to prevent the spread of the threat. Required argument. This argument must be specified manually.

`--quarantineObject true|false` specifies whether the malicious object must be deleted when an indicator of compromise is detected. Before deleting an object, Kaspersky Endpoint Security creates a backup copy of it in case you need to restore the object later. Kaspersky Endpoint Security then quarantines the backup copy. Required argument. This argument must be specified manually.

`--scanCriticalAreas true|false` specifies whether the *Critical Areas Scan* task must be run when an indicator of compromise is detected. By default, Kaspersky Endpoint Security checks kernel memory, running processes, and boot sectors. Required argument. This argument must be specified manually.

Example

```
python3 iocScan.py --targetHost 004c7fcd-8940-4d62-8a94-911932dc861b --iocZip
'./ioc.zip' --isolateHost true --scanCriticalAreas true --quarantineObject true
```

quarantineFile.py quarantines a file.

The quarantineFile script automatically creates and runs the *Move file to Quarantine* task when certain events are received in KUMA. *Quarantine* is a special local storage on your computer. A user can quarantine files that the user considers dangerous for the computer. Quarantined files are stored in encrypted form and do not pose a threat to the security of the device.

Limitations

The *Move file to Quarantine* task has the following limitations:

1. The file size may not exceed 100 MB.
2. Files required by Kaspersky Endpoint Security for Linux cannot be deleted:
   - Files that are included in the scope of the Kaspersky Endpoint Security for Linux self-defense technology
   - Files digitally signed by Kaspersky
   - Files included in the trusted zone of the application

Arguments

To configure the *Move file to Quarantine* task, you need to specify the following arguments for the script:

`--targetHost` is the ID of the computer on which the script must be run. Required argument. The script gets this argument from the event.

`--path` is the path to the file. Required argument. The script gets this argument from the event.

`--md5hash` is the MD5 hash of the file. The script gets this argument from the event.

`--sha256hash` is the SHA256 hash of the file. The script gets this argument from the event.

Examples

```
python3 quarantineFile.py --targetHost 004c7fcd-8940-4d62-8a94-911932dc861b --path
'C:\Folder\File.txt'

python3 quarantineFile.py --targetHost 004c7fcd-8940-4d62-8a94-911932dc861b --path
'C:\Folder\' --sha256hash 12345678910123456789012345678901 2
```

deleteFile.py deletes a file.

The deleteFile script automatically creates and runs the *Delete file* task when certain KUMA events are detected. For example, you can remotely delete a file when responding to threats.

Limitations

Files required by Kaspersky Endpoint Security for Linux cannot be deleted:

- Files that are included in the scope of the Kaspersky Endpoint Security for Linux self-defense technology
- Files digitally signed by Kaspersky
- Files included in the trusted zone of the application

Arguments

To configure the *Delete file* task, you need to specify the following arguments for the script:

`--targetHost` is the ID of the computer on which the script must be run. Required argument. The script gets this argument from the event.

`--searchInSubfolders true|false` specifies whether subfolders must be traversed in search for the file. Required argument. This argument must be specified manually.

`--path` is the path to the file. Required argument. The script gets this argument from the event.

`--md5hash` is the MD5 hash of the file. The script gets this argument from the event.

`--sha256hash` is the SHA256 hash of the file. The script gets this argument from the event.

Examples

```
python3 deleteFile.py --targetHost 004c7fcd-8940-4d62-8a94-911932dc861b --path
'C:\Folder\File.txt' --searchInFolders false
```

```
python3 deleteFile.py --targetHost 004c7fcd-8940-4d62-8a94-911932dc861b --path '1' -
-searchInFolders false --sha256hash 12345678910123456789012345678901 2
```

getFile.py gets a file.

The getFile script automatically creates and runs the *Get file* task when certain events are received in KUMA. For example, you can configure it to fetch an event log file that a third-party application creates. The task saves a copy of the file in the quarantine. *Quarantine* is a special local storage on your computer. You can use the Web Console to download this file to your computer from the quarantine (**Operations** → **Storages** → **Quarantine**). On the user's computer, the file remains in its original folder.

Limitation

The file size may not exceed 100 MB.

Arguments

To configure the *Get file* task, you need to specify the following arguments for the script:

`--targetHost` is the ID of the computer on which the script must be run. Required argument. The script gets this argument from the event.

`--path` is the path to the file. Required argument. The script gets this argument from the event.

`--md5hash` is the MD5 hash of the file. The script gets this argument from the event.

`--sha256hash` is the SHA256 hash of the file. The script gets this argument from the event.

Examples

```
python3 getFile.py --targetHost 004c7fcd-8940-4d62-8a94-911932dc861b --path
'C:\Folder\File.txt'

python3 getFile.py --targetHost 004c7fcd-8940-4d62-8a94-911932dc861b --path
'C:\Folder\' --md5hash 12345678910123456789012345678901 2
```

isolateHost.py isolates the computer from the network.

The isolateHost script automatically isolates a computer from the network when certain KUMA events are detected.

Once Network Isolation is enabled, the application terminates all active TCP/IP network connections and blocks all new connections on the computer, except for the following connections:

- Connections specified in exclusions from Network Isolation
- Connections initiated by Kaspersky Endpoint Security services
- Connections initiated by the Kaspersky Security Center Network Agent
- Connections to the SVM and Integration Server if the application is being used in Light Agent mode
- Connections to KSN when the application is integrated with Kaspersky Managed Detection and Response

Arguments

To configure Network Isolation, you need to specify the following arguments for the script:

`--targetHost` is the ID of the computer on which the script must be run. Required argument. The script gets this argument from the event.

`--action` is the Network Isolation mode. Required argument. This argument must be specified manually. Possible values:

- `1` isolates the computer from the network.
- `0` disables Network Isolation.

Example

```
python3 isolateHost.py --targetHost 004c7fcd-8940-4d62-8a94-911932dc861b --action 1
```

preventExecution.py prevents execution of objects.

The preventExecution script automatically prevents objects from running when certain KUMA events are detected. Execution prevention for objects lets you control the running of executable files and scripts, as well as the opening of office format files. In this way, you can, for example, prevent the launch of applications that you consider unsafe. As a result, the spread of the threat may be stopped. Execution prevention for objects supports a certain set of office file extensions and a certain set of script interpreters. An *execution prevention rule* is a set of criteria that an application takes into account when responding to an object starting, such as when blocking an object from running. The application identifies files by their path or checksum using MD5 and SHA256 hashing algorithms.

> Kaspersky Endpoint Security monitors the creation of new execution prevention rules and ignores duplicate rules. That is, if the launch of an object is already prevented, the application does not create a new rule. For this to be true, all of the parameters of the rule must match (path to the file and file hash), and the rule must be enabled.

Limitations

Execution prevention for objects has the following limitations:

1. Prevention rules do not apply to files on CDs or ISO images. The application does not block the running or opening of such files.
2. A prevention rule cannot be applied to files required by Kaspersky Endpoint Security for Linux:
   - Files that are included in the scope of the Kaspersky Endpoint Security for Linux self-defense technology
   - Files digitally signed by Kaspersky
   - Files included in the trusted zone of the application
3. We do not recommended creating more than 5000 execution prevention rules because this may make the system unstable.

Arguments

To configure Execution prevention for objects, you need to specify the following arguments for the script:

`--targetHost` is the ID of the computer on which the script must be run. Required argument. The script gets this argument from the event.

`--hash` is the MD5 hash of the file. Required argument. The script gets this argument from the event.

`--path` is the full path to the file, or the path to the file if you enter the file hash. Required argument. The script gets this argument from the event.

Example

```
python3 preventExecution.py --targetHost '422db51e-9383-46d4-b75d-f4d1d20fff2c' --
path 'C:\test5\run10.exe' --hash '8806b7f358884a049675b42c5f75ba10'
```

startProcess.py starts a process.

The startProcess script automatically creates and runs the *Start process* task when certain KUMA events are detected. For example, you can remotely run a utility that creates a report about the computer's configuration. You can then use the *Get file* task to receive the created report file in the Kaspersky Security Center Web Console.

Arguments

To configure the *Start process* task, you need to specify the following arguments for the script:

`--targetHost` is the ID of the computer on which the script must be run. Required argument. The script gets this argument from the event.

`--executablePath` is the path to the file. Required argument. You can pass this argument from the event or enter a value manually.

`--arguments` lists additional command line arguments for the process. These arguments must be specified manually.

`--workingFolder` is the path to the working folder of the process. This argument must be specified manually.

Example

```
python3 startProcess.py --targetHost 004c7fcd-8940-4d62-8a94-911932dc861b --
executablePath 'c://a/a.exe'
```

terminateProcess.py terminates a process.

The terminateProcess script automatically creates and runs the *Terminate process* task when certain KUMA events are detected. For example, you can remotely terminate an internet speed test utility that was launched by the *Run process* task.

If you want to prevent a file from running, you can configure the Execution prevention for objects component. You can prevent the execution of executable files, scripts, office format files.

Limitations

Processes of files required by Kaspersky Endpoint Security for Linux cannot be terminated:

- Files that are included in the scope of the Kaspersky Endpoint Security for Linux self-defense technology
- Files digitally signed by Kaspersky
- Files included in the trusted zone of the application

Arguments

To configure the *Terminate process* task, you need to specify the following arguments for the script:

`--targetHost` is the ID of the computer on which the script must be run. Required argument. The script gets this argument from the event.

`--caseSensitive true|false` enables or disables the case-sensitive mode when searching for the file. Required argument. This argument must be specified manually.

`--path` is the path to the file. Required argument. The script gets this argument from the event.

`--md5hash` is the MD5 hash of the file. The script gets this argument from the event.

`--sha256hash` is the SHA256 hash of the file. The script gets this argument from the event.

Examples

```
python3 terminateProcess.py --targetHost 004c7fcd-8940-4d62-8a94-911932dc861b --path
'C:\Folder\file.exe' --caseSensitive true
```

```
python3 terminateProcess.py --targetHost 004c7fcd-8940-4d62-8a94-911932dc861b --path
'C:\Folder\' --caseSensitive false --md5hash 12345678910123456789012345678901 2
```

e. In the **Workers** field, specify the number of workers that the service can run simultaneously to process response rules in parallel. By default, the number of handlers is the same as the number of virtual processors on the server where the service is installed.

f. If necessary, enter an arbitrary description of the response rule in the **Description** field.

g. In the **Filter** field, specify conditions that events must satisfy for the response rule to be applied.

3. Save settings and install the correlator service on the server or restart the active correlator to apply the settings.

When a correlation event with certain fields is received, it triggers the response rule in KUMA, which initiates the EDR action. As a result, a corresponding task is created and run in Kaspersky Security Center. You can view the relevant records in Kaspersky Security Center logs and Kaspersky Endpoint Security logs.

Notification templates

Notification templates are used to create email notifications about events of the following types:

- **Alert created**.
- **Report generated**.
- **Task finished** (only one template of this type can exist).
- **Sources monitoring alert**.
- **KASAP group changed**.

You can create templates of all types of events for the Shared tenant. For all other tenants, only the following event types are available: **Alert created** and **Sources monitoring alert**.

Predefined notification templates.

The notification templates listed in the table below are included in the KUMA distribution kit.

Predefined notification templates.

Template name	Description
[OOTB] New alert in KUMA	Basic notification template.

Creating a notification template

To create a notification template:

1. In the KUMA web interface, open the **Resources** → **Notification templates** section.

   In the right part of the **Resources** section, the **Notification templates** table is displayed.

2. Click the **Add** button and in the displayed window, specify settings as described in the table below.

Notification template settings

Setting	Description
Name	Required setting. Unique name of the template. Must contain 1 to 128 Unicode characters. The name may not begin or end with a space and may not contain control sequences (line feed, etc).
Tenant	Required setting. The name of the tenant that owns the resource. You can create templates of all types of events for the Shared tenant. For all other tenants, only the following types are available: **Alert created** and **Sources monitoring alert**.
Type	Required setting. The type of event for which you want to create a notification template. The following types are available: • **Alert created**. • **Report generated**. • **Task finished**. Only one template of this type can exist. If a template of this type already exists, this type is not available for selection when creating a new template. • **Sources monitoring alert**. • **KASAP group changed**.
Subject	Required setting. Subject of the notification email message. Maximum length: 140 Unicode characters. You can enter the text in English or Russian, depending on the localization selected in the user profile. For each event type, a standard subject text is available when creating a template. You cannot edit the text in the **Subject** field or select a standard text until you select the template type in the **Type** field. In the subject of the message, you can reference the object fields depending on the selected type of event. Example (subject in notification template for an event of the **Alert created** type): `New alert in KUMA: {{.CorrelationRuleName}}`. In place of `{{.CorrelationRuleName}}`, the subject of the notification message will include the name of the correlation rule contained in the `CorrelationRuleName` alert field.

Setting	Description
Template	Required setting. Body of the notification email message. You can enter the text in English or Russian, depending on the localization selected in the user profile. The syntax of the template lets you dynamically populate the notification with data from the specified fields of the corresponding objects. For a description of the notification template syntax and the list of supported fields, see the Notification template syntax article. For each event type, a standard email message subject text is available when creating a template. For convenience, you can open the text of the email message in a separate window by clicking the icon. This opens the **Template** window in which you can edit the text of the notification message. Click **OK** to save the changes and close the window.
Default	Optional setting. Available for the **Task finished** event type. If the check box is selected, the created template is applied instead of the system template to all tasks created after saving the template.
Tags	Optional setting. Tags for the generated notification template. 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.
Description	Optional setting. Arbitrary text describing the template. Maximum length: 4000 Unicode characters.

3. Click **Save**.

The notification template is created and displayed in the **Notification templates** table.

Editing a notification template

To edit a notification template:

1. In the KUMA web interface, open the **Resources** → **Notification templates** section.

   In the right part of the **Resources** section, the **Notification templates** table is displayed.

2. Open the relevant template by clicking its row in the table, and in the displayed window, edit the settings listed in the table in the Creating a notification template article.

3. Click **Save**.

The notification template is modified.

Deleting a notification template

To delete a notification template:

1. In the KUMA web interface, open the **Resources** → **Notification templates** section.

   In the right part of the **Resources** section, the **Notification templates** table is displayed.

2. Select the check boxes next to the notification templates that you want to delete.

   If you want to delete all existing templates, select the check box next to the **Name** column and select **Select all** from the drop-down list.

   If you want to delete all templates on the page, select the check box next to the **Name** column and select **Select all in page** from the drop-down list.

You must select at least one check box.

3. Click the **Delete** button.

   A confirmation window opens. If some of the templates being deleted are linked to other resources or services, these are listed in the confirmation window with a note saying that these templates will not be deleted.

4. Click the **Delete** button to confirm deletion.

Templates that can be safely deleted are deleted.

Notification template syntax

The notification templates use the Go templates syntax. You can use this syntax to dynamically fill notifications with content: load data from the fields of the corresponding objects, insert links, or convert dates to a different format. You can use the template syntax in the subject as well as the body of the message. This article briefly describes the syntax and lists the fields that can be used for certain types of events, and additional functions that can be applied to these fields. You can read more about the syntax in the official Go language documentation.

In the template, you can reference the object fields depending on the selected type of event. The fields of the corresponding objects can contain simple values such as a string or a number:

```
{{ .CorrelationRuleName }}
```

The message will display the alert name, which is the contents of the `CorrelationRuleName` field of the alert data model.

Some fields may contain data arrays, for example, alert fields with related events, assets, and accounts. Such nested objects can be queried by using the `range` function, which sequentially queries the fields of the first 50 nested objects. When using the `range` function to query a field that does not contain a data array, an error is returned. Example:

```
{{ range .Assets }}
Device: {{ .DisplayName }}, creation date: {{ .CreatedAt }}
{{ end }}
```

The message will display the values of the `DeviceHostName` and `CreatedAt` fields from 50 assets related to the alert:

```
Device: <DisplayName field value from asset 1>, creation date: <CreatedAt field value from asset 1>
Device: <DisplayName field value from asset 2>, creation date: <CreatedAt field value from asset 2>
...
// 50 strings total
```

You can use the `limit` function to limit the number of objects returned by the `range` function:

```
{{ range (limit .Assets 5) }}
<strong>Device</strong>: {{ .DisplayName }},
<strong>Creation date</strong>: {{ .CreatedAt }}
{{ end }}
```

The message will display the values of the `DisplayName` and `CreatedAt` fields from 5 assets related to the alert, with the words "Devices" and "Creation date" marked with the `<strong>` HTML tag:

```
<strong>Device</strong>: <DeviceHostName field value from asset 1>,
<strong>Creation date</strong>: <value of the CreatedAt field from asset 1>
<strong>Device</strong>: <DeviceHostName field value from asset N>,
<strong>Creation date</strong>: <CreatedAt field value from asset N>
...
// 10 strings total
```

Nested objects can have their own nested objects. They can be queried by using nested `range` functions:

```
{{ range (limit .Events 5) }}
    {{ range (limit .Event.BaseEvents 10) }}
    Service ID: {{ .ServiceID }}
    {{ end }}
{{ end }}
```

The message will show ten service IDs (`ServiceID` field) from the base events related to five correlation events of the alert (50 lines in total). Please note that events are queried through the nested `EventWrapper` structure, which is located in the `Events` field in the alert. Events are available in the `Event` field of this structure, which is reflected in the example above. Therefore, if field `A` contains nested structure `[B]` and structure `[B]` contains field `C`, which is a string or a number, to query field `C`, you must specify the path `{{ A.C }}`.

Some object fields contain nested dictionaries in key-value format (for example, the `Extra` event field). They can be queried by using the `range` function with the variables passed to it: `range $placeholder1, $placeholder2 := .FieldName`. The values of variables can then be called by specifying their names. Example:

```
{{ range (limit .Events 3) }}
    {{ range (limit .Event.BaseEvents 5) }}
    List of fields in the Extra event field: {{ range $name, $value := .Extra }} {{ $name }} - {{ $value }} <br>
{{ end }}
    {{ end }}
{{ end }}
```

The message will use an HTML tag
 to show key-value pairs from the `Extra` fields of the base events belonging to the correlation events. Data is called from five base events out of each of the three correlation events.

You can use HTML tags in notification templates to create more complex structures. Example table for correlation event fields:

```
<style type="text/css">
  TD, TH {
    padding: 3px;
    border: 1px solid black;
  }
</style>
<table>
  <thead>
    <tr>
        <th>Service name</th>
        <th>Name of the correlation rule</th>
        <th>Device version</th>
    </tr>
  </thead>
  <tbody>
    {{ range .Events }}
    <tr>
        <td>{{ .Event.ServiceName }}</td>
        <td>{{ .Event.CorrelationRuleName }}</td>
        <td>{{ .Event.DeviceVersion }}</td>
    </tr>
    {{ end }}
  </tbody>
</table>
```

Use the **link_alert** function to insert an HTML alert link into the notification email:

```
{{link_alert}}
```

A link to the alert window will be displayed in the message.

Templates also support more complex constructs such as local variable declaration, comparison operators, and control structures. Below is an example of how you can extract the data on max asset category from the alert data and place it in the notifications:

```
{{ $criticalCategoryName := "" }}
{{ $maxCategoryWeight := 0 }}
{{ range .Assets }}
    {{ range .CategoryModels }}
        {{ if gt .Weight $maxCategoryWeight }}
            {{ $maxCategoryWeight = .Weight }}
            {{ $criticalCategoryName = .Name }}
        {{ end }}
    {{ end }}
{{ end }}
{{ if gt $maxCategoryWeight 1 }}
Max asset category: {{ $criticalCategoryName }}
{{ end }}
```

Object fields supported in templates

The following table lists the fields that can be accessed from templates, depending on the type of event.

Supported object fields

Type	Available fields
Alert created	All fields of the alert data model are supported, including nested arrays and structs.
Report generated	`{{ .Tenants }}` —tenants for which the report is created. `{{ .Name }}` —report name. `{{ .StartTime }}` —the beginning of the period for which the report is created, in the "2006-01-02 15:04:05 MSK" format. `{{ .FinishTime }}` —the end of the period for which the report is created, in the "2006-01-02 15:04:05 MSK" format. `{{ .PreviewLink }}` —link for viewing the report. `{{ .DownloadLink }}` —link for downloading the report. Example of adding links to a report notification template: You can `<a href="{{ .PreviewLink }}" target="_blank">` view `</a>` or `<a href="{{ .DownloadLink }}" target="_blank">download</a>` the report.
Task finished	`{{ .Tenants }}` —the tenant in which the task is started. `{{ .TaskKind }}` —the name of the task. `{{ .TaskCreator }}` —the name (login) of the user that created the task. `{{ .StartTime }}` —the start time of the task in the "2006-01-02 15:04:05 MSK" format. `{{ .FinishTime }}` —the end time of the task in the "2006-01-02 15:04:05 MSK" format. `{{ .TaskStatus }}` —task execution status.

Sources monitoring alert	`{{ .SourceTenantName }}` —tenant of the event source. `{{ .SourceDisplayName }}` —name of the source. `{{ .Value }}` —the value that exceeded the bounds of the monitoring policy at the time of the alert creation. `{{ .WebLink }}` —link to the KUMA web interface. `{{ .AlertActiveAt }}` —time at which the alert was active. `{{ .PolicyName }}` —name of the monitoring policy. `{{ .PolicyTenantName }}` —tenant of the monitoring policy. `{{ .PolicyLowerEventBound }}` —lower threshold. `{{ .PolicyUpperEventBound }}` —upper threshold. `{{ .PolicyKind }}` —type of the policy. `{{ .PolicyInterval }}` —interval of the counting period. `{{ .DeviceProduct }}` —source device. `{{ .DeviceHostName }}` —source host name. `{{ .DeviceAddress }}` —MAC address of the source. `{{ .DeviceProcessName }}` —name of the source process.
KASAP group changed	`{{ .UserEmail }}` —display name of the user from Active Directory. `{{ .PreviousGroup}}` —KASAP learning group to which the user belonged at the time when the user's card was opened form an incident or alert. `{{ .CurrentGroup}}` —KASAP training group to which the user was moved from the user's card. `{{ .WebLink }}` —link to the KUMA web interface.

Functions in notification templates

Functions available in templates are listed in the table below.

Functions in templates

Function	Description
`date`	Takes the time in milliseconds (unix time) as the first parameter; the second parameter can be used to pass the time in RFC standard format. The time zone cannot be changed. Example call: `{{ date .FirstSeen "02 Jan 06 15:04" }}` Call result: 18 Nov 2022 13:46 Examples of date formats supported by the function: • `"02 Jan 06 15:04 MST"` • `"02 Jan 06 15:04 -0700"` • `"Monday, 02-Jan-06 15:04:05 MST"` • `"Mon, 02 Jan 2006 15:04:05 MST"` • `"Mon, 02 Jan 2006 15:04:05 -0700"` • `"2006-01-02T15:04:05Z07:00"`
`range`	Lets you iterate over arrays or sets of key-value pairs, sequentially referring to the fields of the first 50 items.

`limit`	This function is called inside the `range` function to limit the list of data. It processes lists that do not have keys, takes any list of data as the first parameter and truncates it based on the second value. For example, the `.Events`, `.Assets`, `.Accounts`, and `.Actions` alert fields can be passed to the function.
	Example call:
	`{{ range (limit .Assets 5) }}`
	`<strong>Device</strong>: {{ .DisplayName }},`
	`<strong>Creation date</strong>: {{ .CreatedAt }}`
	`{{ end }}`

`link_alert`	Generates a link to the alert with the URL specified in the SMTP server connection settings as the KUMA Core server alias or with the real URL of the KUMA Core service if no alias is defined. Example call: `You can find more information about the alert in the KUMA web interface: {{link_alert}}.`

Function	Description

`link_task`	Generates a link to the task in KUMA.
	Example call:
	`Task "{{ .TaskKind }}" created {{ .TaskCreator }} at {{ .StartTime }} and completed at {{ .FinishTime }} with the "{{ .TaskStatus }}" status.`
	`You can find more information about tasks in the KUMA web interface: {{ link_task }}.`

`link`	Takes the form of a link that can be followed. Example call: `{{ link "https://support.kaspersky.com/KUMA/2.1/en-US/233508.htm" }}`

Connectors

Connectors are used for establishing connections between KUMA services and for receiving events actively and passively.

You can specify connector settings on the **Basic settings** and **Advanced settings** tabs. The available settings depend on the selected type of connector.

Connectors can have the following types:

- internal for receiving data from KUMA services using the 'internal' protocol.
- tcp for passively receiving events over TCP when working with Windows and Linux agents.
- udp for passively receiving events over UDP when working with Windows and Linux agents.
- netflow for passively receiving events in the NetFlow format.
- sflow for passively receiving events in the sFlow format. For sFlow, only structures described in sFlow version 5 are supported.
- nats-jetstream for interacting with a NATS message broker when working with Windows and Linux agents.
- kafka for communicating with the Apache Kafka data bus when working with Windows and Linux agents.
- http for receiving events over HTTP when working with Windows and Linux agents.

sql for querying data from a database. KUMA supports multiple types of databases.

Supported database types

The solution 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 of the **sql** type, you must specify general connector settings and individual database connection settings.
- file for getting data from text files when working with Windows and Linux agents. One line of a text file is considered to be one event. `\n` is used as the newline character.
- 1c-log for getting data from 1C technology logs when working with Linux agents. `\n` is used as the newline character. The connector accepts only the first line from a multi-line event record.
- 1c-xml for getting data from 1C registration logs when working with Linux agents. When the connector handles multi-line events, it converts them into single-line events.
- diode for unidirectional data transmission in industrial ICS networks using data diodes.
- ftp for getting data over File Transfer Protocol (FTP) when working with Windows and Linux agents.
- nfs for getting data over Network File System (NFS) when working with Windows and Linux agents.
- wmi for getting data using Windows Management Instrumentation when working with Windows agents.
- wec for getting data using Windows Event Forwarding (WEF) and Windows Event Collector (WEC), or local operating system logs of a Windows host when working with Windows agents.
- etw for getting extended logs of DNS servers.
- snmp for getting data over Simple Network Management Protocol (SNMP) when working with Windows and Linux agents. To process events received over SNMP, you must use the json normalizer. Supported SNMP protocol versions:
  - snmpV1
  - snmpV2
  - snmpV3
- snmp-trap for passively receiving events using SNMP traps when working with Windows and Linux agents. The connector receives snmp-trap events and prepares them for normalization by mapping SNMP object IDs to 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 over SNMP, you must use the json normalizer. Supported SNMP protocol versions:
  - snmpV1
  - snmpV2
- kata/edr for getting KEDR data via the API.
- vmware for getting VMware vCenter data via the API.
- elastic for getting Elasticsearch data. Elasticsearch version 7.0.0 is supported.

- office365 for receiving Microsoft 365 (Office 365) data via the API.
- dfi for receiving Kaspersky Digital Footprint Intelligence data via the API.
- universalAPI for receiveing data from any REST APIs that support JSON responses. Filtering by unique ID and iteration mode are supported.

Some connector types (such as tcp, sql, wmi, wec, and etw) support TLS encryption. KUMA supports TLS 1.2 and 1.3. When TLS mode is enabled for these connectors, the connection is established according to the following algorithm:

- If KUMA is being used as a client:
  1. KUMA sends a connection request to the server with a ClientHello message specifying the maximum supported TLS version (1.3), as well as a list of supported ciphersuites.
  2. The server responds with the preferred TLS version and a ciphersuite.
  3. Depending on the TLS version in the server response:
     - If the server responds to the request with TLS 1.3 or 1.2, KUMA establishes a connection with the server.
     - If the server responds to the request with TLS 1.1, KUMA terminates the connection with the server.
- If KUMA is being used as a server:
  1. The client sends a connection request to KUMA with the maximum supported TLS version, as well as a list of supported ciphersuites.
  2. Depending on the TLS version in the client request:
     - If the ClientHello message of the client request specifies TLS 1.1, KUMA terminates the connection.
     - If the client request specifies TLS 1.2 or 1.3, KUMA responds to the request with the preferred TLS version and a ciphersuite.

Viewing connector settings

To view connector settings:

1. In the KUMA web interface, select **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 KUMA web interface, select **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.

Connector, internal type

Connectors of the **internal** type are used for receiving data from KUMA services using the 'internal' protocol. For example, you must use such a connector to receive the following data:

- 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 type, at the **Event parsing** step, you can pass the name of the file being processed by the collector or the path to the file in the KUMA event field in the **Mapping** table. 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** 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 connector 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.
Type	Connector type: **internal**. Required setting.
Tags	Tags for resource search. Optional setting.
URL	The URL and port that the connector is listening on. You can enter a value in one of the following formats: • `< host name >:< port number >` • `< IPv4 address >:< port number >` • `:< 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`. Only one URL can be added. The resource or service cannot be saved if the value of the field contains spaces or tab characters. Required setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.

Connector, tcp type

Connectors of the **tcp** type are used for passively receiving events over TCP when working with Windows and Linux agents. Settings for a connector of the **tcp** 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.
Type	Connector type: **tcp**. Required setting.
Tags	Tags for resource search. Optional setting.

Setting	Description
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 >` Required setting.
Auditd	This toggle switch enables the auditd mechanism to group auditd event lines received from the connector into an auditd event. If you enable this toggle switch, you cannot select a value in the **Delimiter** drop-down list because **\n** is automatically selected for the mechanism. If you enable this toggle switch in the connector settings of the agent, you need to select **\n** in the **Delimiter** drop-down list in the cor settings of the collector to which the agent sends events. The maximum size of a grouped auditd event is approximately 4,174,304 characters. KUMA classifies Auditd events according to an algorithm. For example, suppose the following records were received for processing: `type=LOGIN msg=audit(1712820601.957:21458): pid=4987 uid=0 subj=0:63:0:0 old-auid=4294967295 auid=0 tty= old-ses=4294967295 ses=2348 res=1` `type=SYSCALL msg=audit(1712820601.957:21458): arch=c000003e syscall=1 success=yes exit=1 a0=7 a1=7ffc9a0 a3=0 items=0 ppid=429 pid=4987 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) comm="cron" exe="/usr/sbin/cron" subj=0:63:0:0 key=(null)` `type=PROCTITLE msg=audit(1712820601.957:21458): proctitle=2F7573722F7362696E2F43524F4E002D66` The algorithm gives one single-line event of the LOGIN type (because the LOGIN type has code 1006 and it is less than 1300, which is AUDIT_FIRST_EVENT), and one multi-line event with SYSCALL and PROCTITLE. When grouping a multi-line auditd event containing records with different prefixes, the resulting event uses the prefix of the record th first. This is necessary for the normalizer to parse the event correctly. For example, suppose the following records were received for p `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit_log: type=PROCTITLE msg=audit(1742441361.524:787467): proctitle=686170726F7879002D57002D6462002D66002F7573722F6C6F63616C2F6574632F686170726F78792F686170726F78` `<182>2025-03-20T06:29:22.643145+03:00 test-server-01 tag_audit_log: type=SOCKADDR msg=audit(1742441361.524:787467): saddr=020026FC0A5E41E80000000000000000` `<182>2025-03-20T06:29:22.643143+03:00 test-server-01 tag_audit_log: type=SYSCALL msg=audit(1742441361.52 arch=c000003e syscall=42 success=no exit=-115 a0=1b a1=55d3f02e8c80 a2=10 a3=55d3eec440d0 items=0 ppid=1 pid=1356 auid=4294967295 uid=99 gid=99 euid=99 suid=99 fsuid=99 egid=99 sgid=99 fsgid=99 tty=(none) ses= comm="haproxy" exe="/usr/local/sbin/haproxy" subj=docker-default key="network_events"<` The `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit_log: type=PROCTITLE msg=audit(...)` arrived first, therefore the resulting event uses the `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit` prefix. When grouping events, the solution ignores strings of the `EOE` type (for example, `type=EOE msg=audit(1633859647.072:120613)` they do not convey information that is important for the events.
Delimiter	• **\n** (used by default) • **\t** • **\0**
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
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.
Character encoding	Character encoding. The default is **UTF-8**.

Setting	Description
Event Buffer TTL	Buffer lifetime for auditd event lines, in milliseconds. Auditd event lines enter the KUMA collector and accumulate in the buffer. This allows multiple auditd event lines to be grouped into a single auditd event. The buffer lifetime countdown begins when the first auditd event line is received or when the previous buffer lifetime expires. Possible values: from 50 to 30,000. The default value is `2000`. This field is available if you have enabled the **Auditd** toggle switch on the **Basic settings** tab. The auditd event lines accumulated in the buffer are kept in the RAM of the server. We recommend caution when increasing the buffer size because memory usage by the KUMA collector may become excessive. You can verify how much server RAM the KUMA collector is using in KUMA metrics. If you want a buffer lifetime to exceed 30,000 milliseconds, we recommend using a different auditd event transport. For example, you can use an agent or pre-accumulate auditd events in a file, and then process this file with the KUMA collector.
Transport header	Regular expression for auditd events, which is used to identify auditd event lines. You can use the default value or edit it. 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 line of the auditd event and discarded for the following lines. You can revert to the default regular expression for auditd events by clicking **Reset to default value**.
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 the solution installation and are stored in the /opt/kaspersky/kuma/core/certificates/ directory of the KUMA Core server. 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 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.

Setting	Description
Compression	Drop-down list for configuring Snappy compression: • **Disabled**. This value is selected by default. • **Use Snappy**.

Connector, udp type

Connectors of the **udp** type are used for passively receiving events over UDP when working with Windows and Linux agents. Settings for a connector 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.
Type	Connector type: **udp**. Required 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 >` Required setting.

Setting	Description
Auditd	This toggle switch enables the auditd mechanism to group auditd event lines received from the connector into an auditd event. If you enable this toggle switch, you cannot select a value in the **Delimiter** drop-down list because **\n** is automatically selected for the mechanism. If you enable this toggle switch in the connector settings of the agent, you need to select **\n** in the **Delimiter** drop-down list in the cor settings of the collector to which the agent sends events. The maximum size of a grouped auditd event is approximately 4,174,304 characters. KUMA classifies Auditd events according to an algorithm. For example, suppose the following records were received for processing: `type=LOGIN msg=audit(1712820601.957:21458): pid=4987 uid=0 subj=0:63:0:0 old-auid=4294967295 auid=0 tty= old-ses=4294967295 ses=2348 res=1` `type=SYSCALL msg=audit(1712820601.957:21458): arch=c000003e syscall=1 success=yes exit=1 a0=7 a1=7ffc9a0 a3=0 items=0 ppid=429 pid=4987 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) comm="cron" exe="/usr/sbin/cron" subj=0:63:0:0 key=(null)` `type=PROCTITLE msg=audit(1712820601.957:21458): proctitle=2F7573722F7362696E2F43524F4E002D66` The algorithm gives one single-line event of the LOGIN type (because the LOGIN type has code 1006 and it is less than 1300, which is AUDIT_FIRST_EVENT), and one multi-line event with SYSCALL and PROCTITLE. When grouping a multi-line auditd event containing records with different prefixes, the resulting event uses the prefix of the record th first. This is necessary for the normalizer to parse the event correctly. For example, suppose the following records were received for p `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit_log: type=PROCTITLE msg=audit(1742441361.524:787467): proctitle=686170726F7879002D57002D6462002D66002F7573722F6C6F63616C2F6574632F686170726F78792F686170726F78` `<182>2025-03-20T06:29:22.643145+03:00 test-server-01 tag_audit_log: type=SOCKADDR msg=audit(1742441361.524:787467): saddr=020026FC0A5E41E80000000000000000` `<182>2025-03-20T06:29:22.643143+03:00 test-server-01 tag_audit_log: type=SYSCALL msg=audit(1742441361.52 arch=c000003e syscall=42 success=no exit=-115 a0=1b a1=55d3f02e8c80 a2=10 a3=55d3eec440d0 items=0 ppid=1 pid=1356 auid=4294967295 uid=99 gid=99 euid=99 suid=99 fsuid=99 egid=99 sgid=99 fsgid=99 tty=(none) ses= comm="haproxy" exe="/usr/local/sbin/haproxy" subj=docker-default key="network_events"<` The `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit_log: type=PROCTITLE msg=audit(...)` arrived first, therefore the resulting event uses the `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit` prefix. When grouping events, the solution ignores strings of the `EOE` type (for example, `type=EOE msg=audit(1633859647.072:120613)`) they do not convey information that is important for the events.
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. This means that the connector does not parse into many, but instead passes it to the normalizer as it is.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
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.
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.
Character encoding	Character encoding. The default is **UTF-8**.

Setting	Description
Event Buffer TTL	Buffer lifetime for auditd event lines, in milliseconds. Auditd event lines enter the KUMA collector and accumulate in the buffer. This allows multiple auditd event lines to be grouped into a single auditd event. The buffer lifetime countdown begins when the first auditd event line is received or when the previous buffer lifetime expires. Possible values: from 50 to 30,000. The default value is `2000`. This field is available if you have enabled the **Auditd** toggle switch on the **Basic settings** tab. The auditd event lines accumulated in the buffer are kept in the RAM of the server. We recommend caution when increasing the buffer size because memory usage by the KUMA collector may become excessive. You can verify how much server RAM the KUMA collector is using in KUMA metrics. If you want a buffer lifetime to exceed 30,000 milliseconds, we recommend using a different auditd event transport. For example, you can use an agent or pre-accumulate auditd events in a file, and then process this file with the KUMA collector.
Transport header	Regular expression for auditd events, which is used to identify auditd event lines. You can use the default value or edit it. 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 line of the auditd event and discarded for the following lines. You can revert to the default regular expression for auditd events by clicking **Reset to default value**.
Compression	Drop-down list for configuring Snappy compression: • **Disabled**. This value is selected by default. • **Use Snappy**.

Connector, netflow type

Connectors of the **netflow** type are used for passively receiving events in the NetFlow format. Settings for a connector of the **netflow** 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.
Type	Connector type: **netflow**. 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.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
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.
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.
Character encoding	Character encoding. The default is **UTF-8**.

Connector, sflow type

Connectors of the **sflow** type are used for passively receiving events in the sFlow format. For sFlow, only structures described in sFlow version 5 are supported. Settings for a connector of the **sflow** 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.
Type	Connector type: **sflow**. Required 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 >` Required setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
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.
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.
Character encoding	Character encoding. The default is **UTF-8**.

Connector, nats-jetstream type

Connectors of the **nats-jetstream** type are used for interacting with a NATS message broker when working with Windows and Linux agents. Settings for a connector of the **nats-jetstream** 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.
Type	Connector type: **nats-jetstream**. 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.
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.
Subject	The topic of NATS messages. Characters are entered in Unicode encoding. Required setting.
GroupID.	The value of the `GroupID` parameter for NATS messages. Maximum length of the value: 255 Unicode characters. The default value is `default`.

Setting	Description
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	The switch enables resource logging. The toggle switch is turned off by default.
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.
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.
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** 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.
	• **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 displayed in the **PFX secret** drop-down list.
	The **Custom PFX** mode is not displayed when **Authorization** is set to **Plain**.

Compression	Drop-down list for configuring Snappy compression: • **Disabled**. This value is selected by default. • **Use Snappy**.

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	The switch enables resource logging. 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.
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.

Setting	Description
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.

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	The switch enables resource logging. The toggle switch is turned off by default.
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. 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 the solution installation and are stored in the /opt/kaspersky/kuma/core/certificates/ directory of the KUMA Core server. 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.

Connector, sql type

Connectors of the **sql** type are used for querying databases. KUMA supports multiple types of databases.

Supported database types

The solution 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 of the **sql** type, you must specify general connector settings and individual database connection settings. Settings for a connector of the **sql** 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.
Type	Connector type: **sql**. Required setting.
Tags	Tags for resource search. Optional setting.
Default query	SQL query that is executed when connecting to the database. Required setting.
Reconnect to DB each time the request is sent	This toggle enables reconnection of the connector to the database every time a query is sent. This toggle switch is turned off by default.
Poll interval, sec	Interval for executing SQL queries in seconds. The default value is 10 seconds.

Setting	Description
Connection	Settings in this group depend on the selected database type for the connection: MySQL Postgres SQLite3 Oracle Firebird ClickHouse ODBC • **Database kind** is the type of the database to connect to. When you select a database type, 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. • The **Secret separately** check box allows viewing the connection information. • **URL** is the connection URL. This lets you 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. When creating connections, if connection information is specified in the URL, strings with credentials containing special characters may not be handled correctly. If an error occurs when creating a connection, but you are sure that the specified 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, [,], :, /, #, %, \. If you select the **Secret separately** check box, the credentials are specified in the secret and are encoded automatically. In this case, you do not need to encode special characters. In the **Use URL from the secret** drop-down list, you can select an existing URL or create a new URL. To create a new URL, select **Create new**. If you want to edit the settings of an existing URL, click the pencil icon next to it. • **Secret** is an urls secret that stores a list of URLs for connecting to the database. This field is displayed if the **Secret separately** check box is selected. • **Identity column** is the name of the column that contains the ID for each row of the table. Required setting. • **Identity seed** is the value in the identity column for determining the row from which you want to start reading data from the SQL table. • **Query** is the additional SQL query that is executed instead of the default SQL query. • **Poll interval, sec** is the SQL query execution interval in seconds. The specified interval is used instead of the default interval for the connector. The default value is 10 seconds. You can add multiple connections or delete a connection. To add a connection, click the **+ Add connection** button. To remove a connection, click the delete ✕ icon next to it.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default 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.

Within a single connector, you can **create a connection** for multiple supported databases. 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 the state file is empty, delete it and restart the collector.

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.

Supported SQL types and their specific usage features

The following SQL types are supported:

- MSSQL.

  For example:

  - `sqlserver://{user}:{password}@{server:port}/{instance_name}?database={database}`

  We recommend using this URL variant.

  - `sqlserver://{user}:{password}@{server}?database={database}`

  The characters `@p1` are used as a placeholder in the SQL query.

  > If you want 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/MariaDB

  For example:

  `mysql://{user}:{password}@tcp({server}:{port})/{database}`

  The characters `?` are used as placeholders in the SQL query.

- PostgreSQL.

  For example: `postgres://{user}:{password}@{server}/{database}?sslmode=disable`

  The characters `$1` are used as a placeholder in the SQL query.

- CockroachDB

  For example:

  `postgres://{user}:{password}@{server}:{port}/{database}?sslmode=disable`

  The characters `$1` are used as a placeholder in the SQL query.

- SQLite3

  For example:

  `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

  Example URL of a secret with the 'oracle' driver:

  ```
  oracle://{user}:{password}@{server}:{port}/{service_name}
  ```

  ```
  oracle://{user}:{password}@{server}:{port}/?SID={SID_VALUE}
  ```

  If the query execution time exceeds 30 seconds, the oracle driver aborts the SQL request, and the following error appears in the collector log: **user requested cancel of current operation**. To increase the execution time of an SQL query, specify the value of the **timeout** parameter in seconds in the connection string, for example:

  ```
  oracle://{user}:{password}@{server}:{port}/{service_name}?timeout=300
  ```

  The `:val` variable is used as a placeholder in the SQL query.

  When querying Oracle DB, if the identity seed is in the datetime format, you must consider the type of the field in the database and, if necessary, add conversions of the time string in the SQL query to make sure the SQL connector works correctly. 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 configuration of the database, 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`. In this case, you need to specify `2021-01-01T00:00:00Z` in the **Identity seed** field, and in the SQL query, perform the conversion 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 WITH TIME ZONE type, then depending on the configuration of the database, 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`). In this case, you need to specify `2021-01-01T00:00:00+03:00` in the Identity seed field, and in the SQL query, perform the conversion 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 details about the `to_timestamp` and `to_timestamp_tz` functions, please refer to the official Oracle documentation.

  To interact with Oracle DB, you must install the libaio1 Astra Linux package.

- Firebird® SQL

  For example:

  ```
  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

  For example:

`clickhouse://{user}:{password}@{server}:{port}/{database}`

A question mark (`?`) is used as a placeholder in the SQL query.

KUMA supports the following data types:

- Data that can be cast to string (such as strings, numeric values, and BLOBs) is displayed as strings.
- Arrays and maps are displayed in JSON format or using the built-in `go fmt.Sprintf("%v",v)` function to display them in the best possible way.

Two methods of connecting to ClickHouse are possible:

- Without credentials, by entering a URL: `clickhouse://host:port/database`
- With credentials, by entering a URL: `clickhouse://user:password@host:port/database`

When using TLS encryption, by default, the connector works only on port 9440. If TLS encryption is not used, by default, the connector works with ClickHouse only on port 9000.

> The connector does not work over HTTP.

If TLS encryption mode is configured on the ClickHouse server, and in connector settings, in the **TLS mode** drop-down list, you have selected **Disabled** or vice versa, the database connection cannot be established.

> The **TLS mode** is used only if the ClickHouse driver is specified.

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 parameters of the **SQL** connector for the ClickHouse connection type, you need to select **Disabled** in the **TLS mode** drop-down list. This value must not be specified if a certificate is used for authentication. If in the **TLS mode** drop-down list, you select **Custom CA**, you need to specify the ID of a secret of the 'certificate' type in the **Secret** field. You also need to select one of the following values in the **Authorization type** drop-down list:

- **Disabled**. If you select this value, you need to leave the **Secret** field blank.
- **Plain**. Select this value if the **Secret separately** check box is selected and the ID of a secret of the 'credentials' type is specified in the **Secret** field.
- **PublicPKI**. Select this value if the **Secret separately** check box is selected and the ID of a secret of the PublicPKI type is specified in the **Secret** field.

If the initial value of the ID contains an indication of time (`datetime`), in the query, you must use a variable for time conversion (`parseDateTimeBestEffort`). For example, if the time is specified as `2021-01-01 00:10:00`, the following query may be used:

```
select connections, username, host, login_time  from connections where login_time >
parseDateTimeBestEffort(?) order by login_time
```

A sequential request for database information is supported in SQL queries. For example, if in the **Query** field, you enter `select * from <name of data table> where id > <placeholder>`, the value of the **Identity seed** field is 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.

We recommend adding the `order by` command to the query string, followed by the sorting field. For example, `select * from table_name where id > ? order by id`.

Examples of SQL requests

SQLite, Firebird, MySQL, MariaDB, ClickHouse: `select * from table_name where id > ? order by id`

MsSQL: `select * from table_name where id > @p1 order by id`

PostgreSQL, Cockroach: `select * from table_name where id > $1 order by id`

Oracle: `select * from table_name where id > :val order by id`

Connector, file type

Connectors of the **file** type are used for getting data from text files when working with Windows and Linux agents. One line of a text file is considered to be one event. `\n` is used as the newline character.

If, while creating the collector, at the Transport step of the **Installation Wizard**, you specified a connector of the file type, at the **Event parsing** step, you can pass the name of the file being processed by the collector or the path to the file in the KUMA event field in the **Mapping** table. 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** connector, the new variables in the normalizer will only work with destinations of the internal type.

To read Windows files, you need to create a connector of the **file** type and manually install the agent on Windows. 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.

For each file that the connector of the **file** type interacts with, a state file (states.ini) is created with the `offset`, `dev`, `inode`, and `filename` parameters. The state file allows the connector, to resume reading from the position where the connector last stopped instead of starting over when rereading the file. Some special considerations are involved in rereading files:

- If the `inode` parameter in the state file changes, the connector rereads the corresponding file from the beginning. When the file is deleting and recreated, the `inode` setting in the associated state file may remain unchanged. In this case, when rereading the file, the connector resumes reading in accordance with the `offset` parameter.
- If the file has been truncated or its size has become smaller, the connector start reading from the beginning.
- If the file has been renamed, when rereading the file, the connector resumes reading from the position where the connector last stopped.
- If the directory with the file has been remounted, when rereading the file, the connector resumes reading from the position where the connector last stopped. You can specify the path to the files with which the connector interacts when configuring the connector in the **File path** field.

Settings for a connector of the **file** 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.
Type	Connector type: **file**. Required setting.
Tags	Tags for resource search. Optional setting.

Setting	Description
Path to the file.	Connectors of the **file** type are used for getting data from text files when working with Windows and Linux agents. One line of a text f considered to be one event. `\n` is used as the newline character. The full path to the file that the connector interacts with. Both Linux and Windows path formats are supported, for example: • Windows: `C:\var\log\test123.log` • Linux: `/var/log/test123.log` Paths that match the following regular expressions 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` In these regular expressions, the double backslashes (\\) are parsed by the system as part of the regular expression syntax and must n manually specified in the path itself. **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 Example: Masks can be used in both Linux and Windows paths, with backslashes replacing forward slashes as appropriate: • Linux: • `/var/log/*som?[1-9].log` • `/mnt/dns_logs/*/dns.log` • `/mnt/proxy/access*.log` • Windows: • `C:\var\log\test123.log` • `D:\data\*\report?.txt` **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/`

Setting	Description

	• `/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 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/`
	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, 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 ar 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 e 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.
	Required setting.

Update timeout, sec	The time in seconds for which the file must not be updated for KUMA to apply the action specified in the **Timeout action** drop-down l file. Default value: `0`, meaning that if the file is not updated, KUMA does not apply any action to it. The entered value must not be less than the value that you entered on the **Advanced settings** in the **Poll interval, ms** field.
Timeout action	The action that KUMA applies to the file after the time specified in the **Update timeout, sec**: • **Do nothing**. The default value. • **Add suffix** adds the .kuma_processed extension to the file name and does not process the file even when it is updated. • **Delete** deletes the file.

Setting	Description
Auditd	This toggle switch enables the auditd mechanism to group auditd event lines received from the connector into an auditd event. If you enable this toggle switch, you cannot select a value in the **Delimiter** drop-down list because **\n** is automatically selected for the mechanism. If you enable this toggle switch in the connector settings of the agent, you need to select **\n** in the **Delimiter** drop-down list in the cor settings of the collector to which the agent sends events. The maximum size of a grouped auditd event is approximately 4,174,304 characters. KUMA classifies Auditd events according to an algorithm. For example, suppose the following records were received for processing: `type=LOGIN msg=audit(1712820601.957:21458): pid=4987 uid=0 subj=0:63:0:0 old-auid=4294967295 auid=0 tty= old-ses=4294967295 ses=2348 res=1` `type=SYSCALL msg=audit(1712820601.957:21458): arch=c000003e syscall=1 success=yes exit=1 a0=7 a1=7ffc9a0 a3=0 items=0 ppid=429 pid=4987 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) comm="cron" exe="/usr/sbin/cron" subj=0:63:0:0 key=(null)` `type=PROCTITLE msg=audit(1712820601.957:21458): proctitle=2F7573722F7362696E2F43524F4E002D66` The algorithm gives one single-line event of the LOGIN type (because the LOGIN type has code 1006 and it is less than 1300, which is AUDIT_FIRST_EVENT), and one multi-line event with SYSCALL and PROCTITLE. When grouping a multi-line auditd event containing records with different prefixes, the resulting event uses the prefix of the record th first. This is necessary for the normalizer to parse the event correctly. For example, suppose the following records were received for p `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit_log: type=PROCTITLE msg=audit(1742441361.524:787467): proctitle=686170726F7879002D57002D6462002D66002F7573722F6C6F63616C2F6574632F686170726F78792F686170726F78` `<182>2025-03-20T06:29:22.643145+03:00 test-server-01 tag_audit_log: type=SOCKADDR msg=audit(1742441361.524:787467): saddr=020026FC0A5E41E80000000000000000` `<182>2025-03-20T06:29:22.643143+03:00 test-server-01 tag_audit_log: type=SYSCALL msg=audit(1742441361.52 arch=c000003e syscall=42 success=no exit=-115 a0=1b a1=55d3f02e8c80 a2=10 a3=55d3eec440d0 items=0 ppid=1 pid=1356 auid=4294967295 uid=99 gid=99 euid=99 suid=99 fsuid=99 egid=99 sgid=99 fsgid=99 tty=(none) ses= comm="haproxy" exe="/usr/local/sbin/haproxy" subj=docker-default key="network_events"<` The `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit_log: type=PROCTITLE msg=audit(...)` arrived first, therefore the resulting event uses the `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit` prefix. When grouping events, the solution ignores strings of the `EOE` type (for example, `type=EOE msg=audit(1633859647.072:120613)`) they do not convey information that is important for the events.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
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.
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.
File/folder polling mode	Specifies how the connector rereads files in the directory: • **Track changes** means the connector rereads files in the directory at an interval in milliseconds specified in the **Poll interval, ms** field if the files are not being updated. The default value. For example, if the files are constantly being updated, and the value of **Poll interval, ms** is `5000`, the connector rereads the files continuously instead of every 5000 milliseconds. If the files are not being updated, the connector rereads them every 5000 milliseconds. • **Track periodic** means the connector rereads files in the directory at an interval in milliseconds specified in the **Poll interval, ms** field, regardless of whether the files are being updated or not.

Setting	Description
Poll interval, ms	The interval in milliseconds at which the connector rereads files in the directory. Default value: `0` means the connector rereads files in the directory every 700 milliseconds. In the **File/folder polling mode** drop-down list, select the mode the connector must use to reread files in the directory. The entered value must not be less than the value that you entered on the **Basic settings** in the **Update timeout, sec** field. We recommend entering a value less than the value that you entered in the **Event Buffer TTL** field because this may adversely affect the performance of Auditd.
Character encoding	Character encoding. The default is **UTF-8**.
Event Buffer TTL	Buffer lifetime for auditd event lines, in milliseconds. Auditd event lines enter the KUMA collector and accumulate in the buffer. This allows multiple auditd event lines to be grouped into a single auditd event. The buffer lifetime countdown begins when the first auditd event line is received or when the previous buffer lifetime expires. Possible values: 700 to 30,000. The default value is `2000`. This field is available if you have enabled the **Auditd** toggle switch on the **Basic settings** tab. The auditd event lines accumulated in the buffer are kept in the RAM of the server. We recommend caution when increasing the buffer size because memory usage by the KUMA collector may become excessive. You can verify how much server RAM the KUMA collector is using in KUMA metrics. If you want a buffer lifetime to exceed 30,000 milliseconds, we recommend using a different auditd event transport. For example, you can use an agent or pre-accumulate auditd events in a file, and then process this file with the KUMA collector.
Transport header	Regular expression for auditd events, which is used to identify auditd event lines. You can use the default value or edit it. 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 line of the auditd event and discarded for the following lines. You can revert to the default regular expression for auditd events by clicking **Reset to default value**.

Connector, 1c-log type

Connectors of the **1c-log** type are used for getting data from 1C technology logs when working with Linux agents. `\n` is used as the newline character. The connector accepts only the first line from a multi-line event record.

Settings for a connector of the **1c-log** 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.
Type	Connector type: **1c-log**. Required setting.
Tags	Tags for resource search. Optional setting.

Setting	Description
Directory path	The full path to the directory with the files that you want 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/` 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/` Required setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

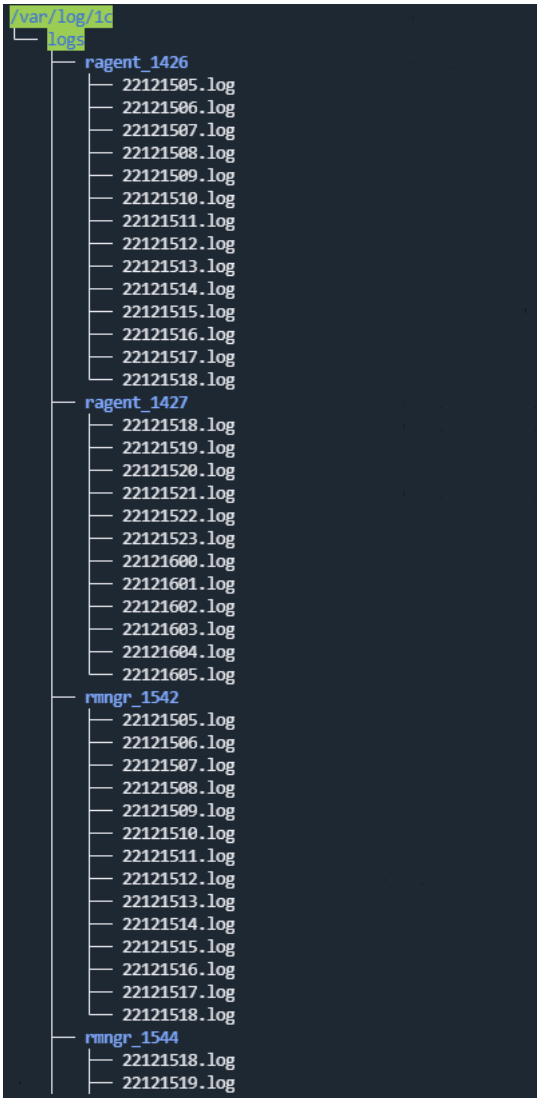
Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Buffer size	Buffer size in bytes for accumulating events in the RAM of the server before sending them for further processing or storage. The value must be a positive integer. Default buffer size: 67,108,864 (64 MB), displayed as 0 in the interface.
File/folder polling mode	Specifies how the connector rereads files in the directory: • **Track changes** means the connector rereads files in the directory at an interval in milliseconds specified in the **Poll interval, ms** field if the files are not being updated. The default value. For example, if the files are constantly being updated, and the value of **Poll interval, ms** is `5000`, the connector rereads the files continuously instead of every 5000 milliseconds. If the files are not being updated, the connector rereads them every 5000 milliseconds. • **Track periodic** means the connector rereads files in the directory at an interval in milliseconds specified in the **Poll interval, ms** field, regardless of whether the files are being updated or not.
Poll interval, ms	The interval in milliseconds at which the connector rereads files in the directory. Default value: `0` means the connector rereads files in the directory every 700 milliseconds. In the **File/folder polling mode** drop-down list, select the mode the connector must use to reread files in the directory.
Character encoding	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.

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 (click the link below to see an image).

Example of a supported 1C technology log structure



Example of a supported 1C technology log structure

- 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 transfer 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 Linux 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

Connector, 1c-xml type

Connectors of the **1c-xml** type are used for getting data from 1C registration logs when working with Linux agents. When the connector handles multi-line events, it converts them into single-line events.

Settings for a connector of the **1c-xml** 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.
Type	Connector type: **1c-xml**. Required setting.
Tags	Tags for resource search. Optional setting.

Setting	Description
Directory path	The full path to the directory with the files that you want 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/` 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/` Required setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
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.
File/folder polling mode	Specifies how the connector rereads files in the directory: • **Track changes** means the connector rereads files in the directory at an interval in milliseconds specified in the **Poll interval, ms** field if the files are not being updated. The default value. For example, if the files are constantly being updated, and the value of **Poll interval, ms** is `5000`, the connector rereads the files continuously instead of every 5000 milliseconds. If the files are not being updated, the connector rereads them every 5000 milliseconds. • **Track periodic** means the connector rereads files in the directory at an interval in milliseconds specified in the **Poll interval, ms** field, regardless of whether the files are being updated or not.
Poll interval, ms	The interval in milliseconds at which the connector rereads files in the directory. Default value: `0` means the connector rereads files in the directory every 700 milliseconds. In the **File/folder polling mode** drop-down list, select the mode the connector must use to reread files in the directory.
Character encoding	Character encoding. The default is **UTF-8**.

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 `DownloadRegistrationLog()` 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 in ascending order of modification time. The connector saves information about the last processed file and does not process files that were modified before it. When the state is reset (for example, when an empty file is found), previously processed files can be processed again.

   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 transfer 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 Linux 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>
```

Example of a processed event

- If there an empty log file (file without events) is found in the directory, the connector resets the information about the last processed file. As a result, previously processed files can be processed again. To avoid processing events twice, use one of the following approaches:
  - Use a file name mask that excludes new empty files from processing, and rename the file after the first event is written to it.
  - Before creating a new file, rename the already-processed files so that they no longer match the connector's search mask.
- 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.

Connector, diode type

Connectors of the **diode** type are used for unidirectional data transmission in industrial ICS networks using data diodes. Settings for a connector of the **diode** 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.
Type	Connector type: **diode**. Required setting.
Tags	Tags for resource search. Optional setting.

Setting	Description
Data diode destination directory	Full path to the directory on the KUMA collector server, into which the data diode moves files with events from the isolated network segment. After the connector has read these files, the files are deleted from the directory. Maximum length of the path: 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/` 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/` Required setting.

Setting	Description
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. You must select the same value in the **Delimiter** drop-down list in the settings of the connector and the destination being used to transmit events from the isolated network segment using a data diode.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. 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.
Poll interval, sec	Interval at which the files are read from the directory containing events from the data diode. The default value is 2 seconds.
Character encoding	Character encoding. The default is **UTF-8**.
Compression	Drop-down list for configuring Snappy compression: • **Disabled**. This value is selected by default. • **Use Snappy**. You must select the same value in the **Snappy** drop-down list in the settings of the connector and the destination being used to transmit events from the isolated network segment using a data diode.

Connector, ftp type

Connectors of the **ftp** type are used for getting data over File Transfer Protocol (FTP) when working with Windows and Linux agents. Settings for a connector of the **ftp** 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.
Type	Connector type: **ftp**. Required setting.
Tags	Tags for resource search. Optional setting.

Setting	Description
URL	URL of file or file mask that begins with the `'ftp://'` schema. You can use `* ? [...]` for the file mask. **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 Example: Masks can be used in both Linux and Windows paths, with backslashes replacing forward slashes as appropriate: • Linux: • `/var/log/*som?[1-9].log` • `/mnt/dns_logs/*/dns.log` • `/mnt/proxy/access*.log` • Windows: • `C:\var\log\test123.log` • `D:\data\*\report?.txt` If the URL does not contain the port number of the FTP server, port `21` is automatically specified. Required setting.
Secret	
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.

Connector, nfs type

Connectors of the **nfs** type are used for getting data over Network File System (NFS) when working with Windows and Linux agents. Settings for a connector of the **nfs** 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.
Type	Connector type: **nfs**. Required setting.

Setting	Description
Tags	Tags for resource search. Optional setting.
URL	Path to the remote directory in the `nfs://< host name >/< path >` format. Required setting.
File name mask	A mask used to filter files containing events. The following wildcards are acceptable "`*`", "`?`", "`[...]`".
Poll interval, sec	Poll interval in seconds. The time interval after which files are re-read from the remote system. The default value is `0`.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.

Connector, wmi type

Connectors of the **wmi** type are used for getting data using Windows Management Instrumentation when working with Windows agents. Settings for a connector of the **wmi** 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.
Type	Connector type: **wmi**. Required setting.
Tags	Tags for resource search. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.
URL	URL of the collector that you created to receive data using Windows Management Instrumentation, for example, `kuma-collector.example.com:7221`. When a collector is created, an agent is automatically created that will get data on the remote device and forward it to the collector service. If you know which server the collector service will be installed on, the URL is known in advance. You can enter the URL of the collector in the **URL** field after completing the installation wizard. To do so, you first need to copy the URL of the collector in the **Resources** → **Active services** section. Required setting.
Default credentials	In the **Remote hosts** table, you need to specify credentials for connecting to hosts that can be used by default. This field is required if the secret for at least one host is not specified in the **Remote hosts** table.

Setting	Description
Remote hosts	Settings of remote Windows devices to connect to. • **Server** is the IP address or name of the device from which you want to receive data, for example, `machine-1`. Cyrillic characters are not supported. Required setting. • **Domain** is the name of the domain in which the remote device resides, for example, `example.com`. Cyrillic characters are not supported. Required setting. • **Log type** are the names of the Windows logs that you want to get. By default, this drop-down list includes only preconfigured logs, but you can add custom log to the list. To do so, enter the names of the custom logs in the **Windows logs** field, then press **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. Required setting. • In the **Secret** field, specify account credentials for accessing a remote Windows device with permissions to read the logs. You can select an existing secret or create a new secret. To create a new secret, select **Create new**. When creating a secret, you do not need to specify the domain for accessing the host in the **User** field because the **Domain** value from the **Remote hosts** table is used. If you want to edit the settings of an existing secret, click the pencil icon. If you leave the **default** value in this field, the credentials specified in the **Default credentials** field are used when connecting to remote Windows devices. **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. You can add multiple remote Windows devices or remove a remote Windows device. To add a remote Windows device, click **+ Add**. To remove a remote Windows device, select the check box next to it and click **Delete**. You can also click the **Import** button to import a file with a list of remote Windows devices in TSV format, which lets you avoid filling in fields manually when you have many such devices. KUMA automatically fills in the data from the file and performs a check. An error is displayed for incorrectly filled fields. You can edit these fields manually and save the settings.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
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.

Setting	Description
Compression	Drop-down list for configuring Snappy compression: • **Disabled**. This value is selected by default. • **Use Snappy**.

If you edit a connector of this type, the **TLS mode** and **Compression** settings are visible and available on the connector resource as well as the collector. If you are using a connector of this type on a collector, the values of **TLS mode** and **Compression** settings are sent to the destination of automatically created agents.

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

Connector, wec type

Connectors of the **wec** type are used for getting data using Windows Event Forwarding (WEF) and Windows Event Collector (WEC), or local operating system logs of a Windows host when working with Windows agents. Settings for a connector of the **wec** 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.
Type	Connector type: **wec**. Required setting.
Tags	Tags for resource search. Optional setting.

Setting	Description
URL	URL of the collector that you created to receive data using Windows Event Collector, for example, `kuma-collector.example.com:7221`. When a collector is created, an agent is automatically created that will get data on the remote device and forward it to the collector service. If you know which server the collector service will be installed on, the URL is known in advance. You can enter the URL of the collector in the **URL** field after completing the installation wizard. To do so, you first need to copy the URL of the collector in the **Resources** → **Active services** section. Required setting.
Windows logs	The names of the Windows logs that you want to get. By default, this drop-down list includes only preconfigured logs, but you can add custom log to the list. To do so, enter the names of the custom logs in the **Windows logs** field, then press **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. Required setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
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.
Compression	Drop-down list for configuring Snappy compression: • **Disabled**. This value is selected by default. • **Use Snappy**.

If you edit a connector of this type, the **TLS mode** and **Compression** settings are visible and available on the connector resource as well as the collector. If you are using a connector of this type on a collector, the values of **TLS mode** and **Compression** settings are sent to the destination of automatically created agents.

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.

Connector, etw type

Connectors of the **etw** type are used for getting extended logs of DNS servers. Settings for a connector of the **etw** 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.
Type	Connector type: **etw**. Required setting.
Tags	Tags for resource search. Optional setting.
URL	URL of the DNS server. Required setting.
Session name	Session name that corresponds to the ETW provider: Microsoft-Windows-DNSServer {EB79061A-A566-4698-9119-3ED2807060E7}. If in a connector of the **etw** type, the session name is specified incorrectly, an incorrect provider is specified in the session, or an incorrect method is specified for sending events (to send events correctly, on the Windows Server side, you must specify "Real time" or "File and Real time" mode), events will not arrive from the agent, an error will be recorded in the agent log on Windows, and the status of the agent will be green. At the same time, no attempt will be made to get events every 60 seconds. If you modify session settings on the Windows side, you must restart the etw agent and/or the session for the changes to take effect. For details about specifying session settings on the Windows side to receive DNS server events, see the Configuring receipt of DNS server events using the ETW agent section. Required setting.
Extract event information	This toggle switch enables the extraction of the minimum set of event information 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, this toggle switch is enabled and all event data is extracted.
Extract event properties	This toggle switch enables the extraction of event properties. If this toggle switch is disabled, event properties are not extracted, which helps save CPU resources on the machine with the agent. By default, this toggle switch is enabled and event properties are extracted. You can enable the **Extract event properties** switch only if the **Extract event information** toggle switch is enabled.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
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.

Setting	Description
Compression	Drop-down list for configuring Snappy compression: • **Disabled**. This value is selected by default. • **Use Snappy**.

If you edit a connector of this type, the **TLS mode** and **Compression** settings are visible and available on the connector resource as well as the collector. If you are using a connector of this type on a collector, the values of **TLS mode** and **Compression** settings are sent to the destination of automatically created agents.

Connector, snmp type

Connectors of the **snmp** type are used for getting data over Simple Network Management Protocol (SNMP) when working with Windows and Linux agents. To process events received over SNMP, you must use the json normalizer. Supported SNMP protocol versions:

- snmpV1
- snmpV2
- snmpV3

Only one **snmp** connector created in the agent settings can be used in an agent. If you need to use multiple **snmp** connectors, you must create one or all **snmp** connectors as a separate resource and select it in the connection settings.

Available settings for a connector of the **snmp** 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.
Type	Connector type: **snmp**. Required setting.
Tags	Tags for resource search. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Setting	Description
SNMP resource	Settings for connecting to an SNMP resource: • **SNMP version** is the version of the SNMP protocol being used. Required setting. • **Host** is the name or IP address of the host. Possible formats: • `<host name>` • `<IPv4 address>` • `<IPv6 address>` Required setting. • **Port** is the port number to be used when connecting to the host. Typical values are `161` or `162`. Required setting. • **Secret** is the secret that stores the credentials for connecting over the Simple Network Management Protocol. The secret type must match the SNMP version. 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. Required setting. You can add multiple connections to SNMP resources or delete an SNMP resource connection. To create a connection to an SNMP resource, click the **+ SNMP resource** button. To delete a connection to an SNMP resource, click the delete icon next to the SNMP resource.
Settings	Rules for naming the received data, according to which OIDs (object identifiers) are converted to the keys with which the normalizer can interact. Available settings: • **Parameter name** is the name for the data type, for example, `Host name` or `Host uptime`. Required setting. • **OID** is 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`. KUMA requests data using the SNMP GET command. Required setting. • **Key** is a unique identifier returned in response to a request to the device with the value of the requested parameter, for example, `sysName`. You can reference the key when normalizing data. Required setting. • If the **MAC address** check box is selected, 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. You can do the following with rules: • Add multiple rules. To add a rule, click the **+ Add** button. • Delete rules. To delete a rule, select the check box next to it and click **Delete**. • Clear rule settings. To do so, click the **Clear all values** button.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.

Connector, snmp-trap type

Connectors of the **snmp-trap** type are used for passively receiving events using SNMP traps when working with Windows and Linux agents. The connector receives snmp-trap events and prepares them for normalization by mapping SNMP object IDs to 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 over SNMP, you must use the json normalizer. Supported SNMP protocol versions:

- snmpV1
- snmpV2

Settings for a connector of the **snmp-trap** 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.
Type	Connector type: **snmp-trap**. Required setting.
Tags	Tags for resource search. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.
SNMP resource	Connection settings for receiving snmp-trap events: • **SNMP version** is the version of the SNMP protocol being used: • **snmpV1** • **snmpV2** For example, Windows uses the snmpV2 version of the SNMP protocol by default. Required setting. • **URL** is the URL for receiving SNMP trap events. 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>` Required setting. You can add multiple connections or delete a connection. To add a connection, click the **+ SNMP resource** button. To remove a SNMP resource, click the delete ✕ icon next to it.

Setting	Description
Settings	Rules for naming the received data, according to which OIDs (object identifiers) are converted to the keys with which the normalizer can interact. Available settings: • **Parameter name** is the name for the data type, for example, `Host name` or `Host uptime`. Required setting. • **OID** is 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`. KUMA requests data using the SNMP GET command. Required setting. • **Key** is a unique identifier returned in response to a request to the device with the value of the requested parameter, for example, `sysName`. You can reference the key when normalizing data. Required setting. • If the **MAC address** check box is selected, 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. You can do the following with rules: • Add multiple rules. To add a rule, click the **+ Add** button. • Delete rules. To delete a rule, select the check box next to it and click **Delete**. • Clear rule settings. To do so, click the **Clear all values** button. • Populate the table with mappings for OID values received in WinEventLog logs. To do this, click the **Apply OIDs for WinEventLog** button. If more data needs to be determined and normalized in the incoming events, add to the table rows containing OID objects and their keys. 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.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default 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.

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, use 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.

Connector, kata/edr type

Connectors of the **kata/edr** type are used for getting KEDR data via the API. Settings for a connector of the **kata/edr** 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.
Type	Connector type: **kata/edr**. 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.
Secret	Secret that stores the credentials for connecting to the KATA/EDR server. 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. Required setting.
External ID	Identifier for external systems. KUMA automatically generates an ID and populates this field with it.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. We only recommend configuring a conversion if you find invalid characters in the fields of the normalized event. By default, no value is selected.
Number of events	Maximum number of events in one request. By default, the value set on the KATA/EDR server is used.

Setting	Description
Events fetch timeout	The time in seconds to wait for receipt of events from the KATA/EDR server. Default value: `0`, which means that the value set on the KATA/EDR server is used.
Client timeout	Time in seconds to wait for a response from the KATA/EDR server. Default value: `0`, corresponding to 1800 seconds.
KEDRQL filter	Filter of requests to the KATA/EDR server. For more details on the query language, please refer to the KEDR Help.

Connector, vmware type

Connectors of the **vmware** type are used for getting VMware vCenter data via the API. Settings for a connector of the **vmware** 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.
Type	Connector type: **vmware**. Required setting.
Tags	Tags for resource search. Optional setting.
URL	URL of the VMware API. You need to include the hostname and port number in the URL. You can only specify one URL. Required setting.
VMware credentials	Secret that stores the user name and password for connecting to the VMware API. 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. Required setting.
Sleep time, sec	Time to wait after a request that did not return events before making a new request. The default value is 5 seconds. If you specify `0`, the default value is used.
Fetch size	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. By default, events are read from the VMware API from the time when the collector was started. If started after the collector is stopped, the events are read from the last saved date.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
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.

Connector, elastic type

Connectors of the **elastic** type are used for getting Elasticsearch data. Elasticsearch version 7.0.0 is supported. Settings for a connector of the **elastic** 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.
Type	Connector type: **elastic**. Required setting.

Setting	Description
Tags	Tags for resource search. Optional setting.
Connection	Elasticsearch server connection settings: • **URL** is the URL of the Elasticsearch server. 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. • **Index name** is the name of the index in Elasticsearch. Required setting. • **Query** is the Elasticsearch query. We recommend specifying the `size` parameter in the query to prevent performance problems with KUMA and Elasticsearch, as well as the `sort` parameter for the sorting order. The following values are possible for the `sort` parameter in the query: `asc`, `desc`, or a custom sorting order by specific fields in accordance with the Elasticsearch syntax. To sort by a specific field, we recommend also specifying the `"missing" : "_first"` parameter next to the `"order"` parameter to prevent errors in cases when this field is absent in any document. For example, `"sort": { "DestinationDnsDomain.keyword": {"order": "desc", "missing" : "_first" } }`. For more details on sorting, please refer to the Elasticsearch documentation. Query example: `"query" : { "match_all" : {} }, "size" : 25, "sort": {"_doc" : "asc"}` Required setting. • **Elastic credentials** is the secret that stores the credentials for connecting to the Elasticsearch server. 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. • **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. 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. • **Poll interval, sec** is the 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. You can add multiple connections to an Elasticsearch server resources or delete an Elasticsearch server connection. To add an Elasticsearch server connection, click the **+ Add connection** button. To delete an Elasticsearch server connection, click the delete ✕ icon next to the Elasticsearch server.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.

Connector, office365 type

Connectors of the **office365** type are used for receiving Microsoft 365 (Office 365) data via the API.

Available settings for a connector of the **office365** 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.
Type	Connector type: **office365**. Required setting.
Tags	Tags for resource search. Optional setting.
Office365 content types	Content types that you want to receive in KUMA. The following content types are available, providing information about actions and events in Microsoft 365, grouped by information source: • **Audit.General** • **Audit.AzureActiveDirectory** • **Audit.Exchange** • **Audit.Sharepoint** • **DLP.All** You can find detailed information about the properties of the available content types and related events in the schema on the Microsoft website. Required setting. You can select one or more content types.
Office365 tenant ID	Unique ID that you get after registering an account with Microsoft 365. If you do not have one, contact your administrator or Microsoft. Required setting.
Office365 client ID	Unique ID that you get after registering an account with Microsoft 365. If you do not have one, contact your administrator or Microsoft. Required setting.
Authorization	Authorization method for connecting to Microsoft 365. The following authorization methods are available: • **PFX**. Using a PFX secret. • **Token**. Using a 'token' secret. For more information, see the section on secrets.
Office365 credentials	The field becomes available after selecting the authorization method. You can select one of the available authorization secrets or create a new secret of the selected type. Required setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.
Authentication endpoint	The URL that is used for connection and authorization. By default, a connection is made to https://login.microsoftonline.com.
Resource endpoint	URL from which the events are to be received. The default address is https://manage.office.com.
Look back interval, hours	The period for which all new events are requested, in hours. To avoid losing some events, it is important to set overlapping event reception intervals, because some types of Microsoft 365 content may be sent with a delay. In this case, previously received events are not duplicated. By default, all new events for the last 12 hours are requested.
Request timeout, sec	Time to wait for a response to a request to get new events, in seconds. The default response timeout is 30 seconds.

Setting	Description
Retry interval, sec	The time in seconds after which a failed request to get new events must be repeated. By default, a request to get new events is repeated 10 seconds after getting an error or no response within the specified timeout.
Clean interval, sec	How often obsolete data is deleted, in seconds. The minimum value is 300 seconds. By default, obsolete data is deleted every 1800 seconds.
Poll interval, min	How often requests for new events are sent, in minutes. By default, requests are sent every 10 minutes.
Proxy server	Proxy settings, if necessary to connect to Microsoft 365. You can select one of the available proxy servers or create a new proxy server.

Connector, dfi type

Connectors of the **dfi** are used for receiving Kaspersky Digital Footprint Intelligence data via the API. Settings for a connector of the **dfi** 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.
Type	Connector type: **dfi**. Required setting.
Tags	Tags for resource search. Optional setting.

URL	URL of the Kaspersky Threat Intelligence Portal service API. In the URL, you need to include the hostname and port number, or just the hostname. You can only specify one URL. Kaspersky TIP address: https://tip.kaspersky.com/api.
	The following URL formats are supported:
	• `host name : port` or `host name`
	Examples:
	`http://tip_hostname:2134/api`
	`http://tip_hostname/api`
	• `IPv4 address : port` or `IPv4 address`
	Examples:
	`http://127.0.0.1:2134/api`
	`http://127.0.0.1/api`
	• `[ IPv6 address ]: port` or `[ IPv6 address ]`.
	The IPv6 address must be specified in square brackets. Examples:
	`http://[1050:0:0:0:5:600:300c:326b]:2134/api`
	`http://[1050:0:0:0:5:600:300c:326b]/api`.
	KUMA does not allow saving a resource or service if the URL field contains a tab or space character.
	Required setting.

Setting	Description
Secret	Secret that stores the credentials for connecting to the Kaspersky Threat Intelligence Portal server. 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 **Token** field, specify the API token for authentication on the Kaspersky Threat Intelligence Portal. You can find detailed information on getting an API token in the Kaspersky Threat Intelligence Portal Help. 3. If necessary, in the **Tags** field, add tags for the secret. 4. If necessary, enter a description of the secret in the **Description** field. 5. Click the **Create** button. The secret is added and displayed in the **Secret** drop-down list. Required setting.
Poll interval	The interval in seconds between requests to the Kaspersky Digital Footprint Intelligence service. Enter a number from 60 to 3600 (that is, from 1 minute to 60 minutes). The default value is `120`. Required setting.
Start timestamp	The starting date and time for the data that the Kaspersky Digital Footprint Intelligence service generates in response to the request. By default, a request to Kaspersky Digital Footprint Intelligence is generated with data for the last 30 days up to the moment when the collector was started. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Proxy server	Proxy server settings, if one is necessary to connect to the Kaspersky Threat Intelligence Portal server. You can select one of the available proxy servers or create a new proxy server. By default, a proxy server is not used.
Character encoding	Character encoding. The default is **UTF-8**.

Connector, 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** → **Active services** section with a yellow status.

The **universalAPI** connector does not replace other existing API connectors, such as **office365**.

Settings for a connector of the **universalAPI** 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.
Type	Connector type: **universalAPI**. Required setting.
Tags	Tags for resource search. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.
URL	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.
Poll interval, sec	The poll interval for the specified URL, in seconds. Enter a number from `60` to `86,400` (that is, from 1 minute to 24 hours). The default value is `3600`.
HTTP method	HTTP method used for API requests. Available values: • **GET** • **POST** • **PUT** • **DELETE** Required setting.
Authorization	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.
Secret ID	This field is available if the **Authorization** field is set to **Plain**. You need to select a secret of the **credentials** type. Required setting.
Token	This field is available if the **Authorization** field is set to **Token**. You need to select a secret of the **token** type. Required setting.
Token name	This field is available if the **Authorization** field is set to **Token**. Available values: • **Bearer** • **OAuth** You can also specify the name manually. Required setting.
Request body	The body of the HTTP request. You can also use variables: `$token`, `$username`, `$password`, `$identityValue`.

Setting	Description
Path to data in JSONPath format	The data path in the structure of the API response that points to an object or a list of objects. The path is specified in JSONPath format (for example, `$.data`, `$.data.edges[0].node.event`, `$.data.edges[*].node.event`). Required setting.
Identity field	Toggle switch for enabling a unique ID by which events will be selected. The toggle switch is turned off by default.
Path to the field in JSONPath format	This field is available if the **Identity field** toggle switch is enabled. The path to the ID field in the structure of the API response element is specified in the JSONPath format (for example, `$.id`). Required setting.
Type	This 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.
Date format	This 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., `$identityValue` in `queryParams`, `headers`, `body`). If the date format includes a time zone, the string is generated in the server's time zone. If the date format does not include a time zone, the string is generated as UTC.
Identity seed	This field is available if the **Identity field** toggle switch is enabled. The identity seed is specified as a string in the format corresponding to the selected **Type**. For the **date** type, you need to specify the value in the format specified in the **Date format** field. This value can contain Unicode characters. The minimum length is 1 character.
Iteration parameters	Toggle switch that enables the iteration mode used for APIs with staged data output. The toggle switch is turned off by default.
Path to the field with the iteration key in JSONPath format	The path to the iteration key field in the structure of the API response element in JSONPath format. Required setting.
Poll interval in iteration mode (sec.)	URL poll interval in iteration mode, in seconds. Maximum value: `86,400` (24 hours). The default value is `60`.
Iteration key URL parameter	Name of the URL parameter in which the iteration key is passed when making a request to the API.
Optional request parameters	Additional request parameters in the `map[string][]string` format. You can also use variables: `$token`, `$username`, `$password`, `$identityValue`. To add a parameter, click the **+ Add** button and fill in the **Key** and **Value** fields. To delete a parameter, select the check box next to it and click **Delete**. If iteration is enabled and the state has an iteration key, it is automatically substituted into the request parameters.
Request headers	Optional HTTP request headers in the `map[string][]string` format. You can also use variables: `$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	Additional fields 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

Setting	Description
Debug	Toggle switch that enables debug mode for the connector. Turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.
Proxy server	Proxy settings, if necessary to connect to the API. You can select one of the available proxy servers or create a new proxy server.

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.
	• **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 **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 (sec)	Time to wait for a response to an API request, in seconds. The default response timeout is 30 seconds.
Retries	The number of repeated attempts to make a request if an error occurs. The default value is `1`.
Retry interval (sec.)	Interval between retries, in seconds. The default value is `0`.

Predefined connectors

The KUMA distribution kit includes the predefined connectors listed in the table below.

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 Person Monitor SQL	Collects system audit events from the MS SQL DBMS of InfoWatch Person Monitor 10.2.
[OOTB] InfoWatch Traffic 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] Kaspersky Security for MS Exchange Server SQL	Collects events from the MS SQL DBMS of Kaspersky Security for Microsoft Exchange Servers version 9.0. 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 system. 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 system. 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 system. 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] Parsec ParsecNET MSSQL	Collects events from the MS SQL DBMS of ParsecNET 3. 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.
[OOTB] TeamPass MySQL	Collects events from the MySQL DBMS of the TeamPass system version 3.1.
[OOTB] Zabbix MySQL	Collects events from the MySQL DBMS of Zabbix version 6.4.

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.

A secret created inside a resource in the Shared tenant is not displayed in the **Resources and services → Secrets** section and is not viewable, even by a user with the General Administrator role. Such a secret can be viewed and manipulated only within the resource in which it was created.

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.

    > The certificate included in the PFX file cannot be encrypted with the AES-128-CBC algorithm. KUMA does not support such certificates.

  - **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 value of **Community** (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.

    Additional settings depending on the selected level may be displayed.

  - **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.

- **fingerprint**—this type of secret is used to store the **Elastic fingerprint** value that can be used when connecting to the Elasticsearch server.

- **PublicPKI**—this type of secret is used to connect a KUMA collector to ClickHouse. If you select this option, you must specify the secret containing the base64-encoded PEM private key and the public key.

Predefined secrets

The secrets listed in the table below are included in the KUMA distribution kit.

Predefined secrets

Secret name	Description
[OOTB] Continent SQL connection	Stores confidential data and settings for connecting to the APKSh Kontinent DB system. To use it, you must specify the login name and password of the database.
[OOTB] InfoWatch Person Monitor MS SQL connection	Stores confidential data and settings for connecting to the MS SQL DB of the InfoWatch Person Monitor system. To use it, you must specify the login name and password of the database.
[OOTB] InfoWatch Traffic Monitor SQL Connection	Stores confidential data and settings for connecting to the database of the InfoWatch Traffic Monitor system. To use it, you must specify the login name and password of the database.
[OOTB] Kaspersky Security for Exchange SQL connection	Stores confidential data and settings for connecting to the database of the Kaspersky Security for Microsoft Exchange Servers system. 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 DB of Kaspersky Security Center. 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 DB of Kaspersky Security Center. To use it, you must specify the login name and password of the database.
[OOTB] KSC PostgreSQL connection	Stores confidential data and settings for connecting to the PostgreSQL DB of Kaspersky Security Center. 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 system DB. To use it, you must specify the login name and password of the database.
[OOTB] Parsec MSSQL connection	Stores confidential data and settings for connecting to the MSSQL DB of the Parsec system. 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.
[OOTB] TeamPass MySQL connection	Stores confidential data and settings for connecting to the MySQL DB of the TeamPass system. To use it, you must specify the login name and password of the database.
[OOTB] Zabbix audit MySQL	Stores confidential data and settings for connecting to the MySQL DB of the Zabbix system. To use it, you must specify the login name and password of the database.

Segmentation rules

In KUMA, you can configure *alert segmentation rules*, that is, the rules for dividing similar correlation events into different alerts.

By default, if a correlation rule is triggered several times in the correlator, all correlation events created as a result of the rule triggering are attached to the same alert. Alert segmentation rules allow you to define the conditions under which different alerts are created based on the correlation events of the same type. This can be useful, for example, to divide the stream of correlation events by the number of events or to combine several events having an important distinguishing feature into a separate alert.

Alert segmentation is configured in two stages:

1. *Segmentation rules* are created. They define the conditions for dividing the stream of correlation events.
2. Segmentation rules are linked to the correlation rules within which they must be triggered.

Creating a segmentation rule

Alert segmentation rules are used to divide correlation events of the same type among different alerts. Conditions for correlation events for which separate alerts must be generated are specified in the segmentation rules.

To create a segmentation rule:

1. In the KUMA web interface, go to **Resources** → **Segmentation rules**, select a tenant and click the **Create new** button.

   This opens the **Create segmentation rule** window.

2. In the **Name** field, enter a name for the segmentation rule. The name must be unique and contain 1 to 128 Unicode characters.

3. If necessary, in the **Tenant** drop-down list, change the tenant to which the segmentation rule belongs.

4. In the **Type** drop-down list, specify the type of the segmentation rule. Available values:

   - **By filter**—an alert is created if the correlation events match the filter conditions specified in the **Filter** group of settings.

     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**. 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. You can use operands and operators to specify conditions:

     - Left operand and Right operand—used to specify the values to be processed by the operator.

       The left operand contains the names of the event fields that are processed by the filter.

       For the right-hand operand, you can select the type of the value: **constant** or **list**, and specify the value.

     - **Available operators**

       - **=**—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.

You can also add a filter in code mode. To do this, go to the **Code** tab in the **Filter** section and specify the filter in the input field. When entering parameter values, autocompletion is supported in the form of a list of hints. To display the list of all hints in the field, press **Ctrl**+**Space**.

> Note that the code mode allows you to use all possible values for rules in KUMA in the list of hints. Some of these values may not be valid for segmentation rules (for example, `activelist`, `dictionary`, `hasbit`).

- **By identical fields**—an alert is created if the correlation event contains the event fields specified in the **Correlation rule identical fields** drop-down list.

  You can select identical fields from the **Correlation rule identical fields** field by selecting check boxes. You can remove added fields by clicking the X icon next to the name of the identical field.

  Example of grouping fields usage

  A rule that detects a network scan generates only one alert, even if there are multiple devices that scan the network. If you create an alert segmentation rule based on the `SourceAddress` event grouping field and then bind this segmentation rule to a correlation rule, alerts are created for each address from which a scan is performed when the rule is triggered.

  In this example, if the correlation rule name is "`Network. Possible port scan`", and the "`from {{.SourceAddress}}`" value is specified as the alert naming template in the segmentation rule resource, alerts are created that look like this:

  - `Network. Possible port scan (from 10.20.20.20)`
  - `Network. Possible port scan (from 10.10.10.10)`

- **By event limit**—an alert is created if the number of correlation events in the previous alert exceeds the value specified in the **Correlation events limit** field.

5. In the **Alert naming template** field, specify the template to be used for naming alerts generated by this segmentation rule. The default value is `{{.Timestamp}}`. In the template field, you can specify text, as well as event fields in the `{{.<Event field name>}}` format. When generating the alert name, the value of the event field is substituted instead of the event field name. The name of the alert created using the segmentation rules has the following format: `<Name of the correlation rule that created the alert> <text from the alert naming template field> <value of the specified event field>`. For example, if the triggered correlation rule name is named "`Connection. Insecure port usage`", and the segmentation rule name template contains `{{.Timestamp}}`, alerts of the following form are created:

   - `Connection. Insecure port usage (Tue, 29 Jul 2025 07:56:43 UTC)`
   - `Connection. Insecure port usage (Tue, 29 Jul 2025 07:59:46 UTC)`

   If multiple segmentation rules are executed with different alert naming templates, they are listed in the alert name separated by the "`|`" character.

6. If necessary, in the **Tags** drop-down list, select the tags for the event segmentation rule.

7. If necessary, enter a description of the segmentation rule in the **Description** field. You can enter up to 4,000 Unicode characters.

The segmentation rule is created. Next, you need to link the segmentation rule to a correlation rule.

Linking segmentation rules to correlation rules

Links between a segmentation rule and correlation rules are created separately for each tenant. They are displayed in the **Settings** → **Alerts** → **Segmentation** section of the KUMA web interface in the table with the following columns:

- **Tenant**—the name of the tenant that owns the segmentation rules.
- **Updated**—date and time of the last update of the segmentation rules.
- **Disabled**—this column displays a label if the segmentation rules are turned off.

To link an alert segmentation rule to the correlation rules:

1. In the KUMA web interface, open the **Settings** → **Alerts** → **Segmentation** section.
2. Select the tenant for which you would like to create a segmentation rule:
   - If the tenant already has segmentation rules, select it in the table.
   - If the tenant has no segmentation rules, click **Add settings for a new tenant** and select the relevant tenant from the **Tenant** drop-down list.

   A table with the created links between segmentation and correlation rule is displayed.
3. In the **Segmentation rule links** group of settings, click **Add** and specify the segmentation rule settings:
   - **Name** (required)—specify the segmentation rule name in this field. Must contain 1 to 128 Unicode characters.
   - **Tenants and correlation rules** (required)—in this drop-down list, select the tenant and its correlation rule to separate the events of this tenant into an individual alert. You can select several correlation rules.
   - **Segmentation rule** (required)—in this group of settings, select a previously created segmentation rule that defines the segmentation conditions.
   - **Disabled**—select this check box to disable the segmentation rule link.
4. Click **Save**.

The segmentation rule is linked to the correlation rules. Correlation events created by the specified correlation rules are combined into a separate alert with the name defined in the segmentation rule.

To disable links between segmentation rules and correlation rules for a tenant:

1. Open the **Settings** → **Alerts** section of the KUMA web interface and select the tenant whose segmentation rules you want to disable.
2. Select the **Disabled** check box.
3. Click **Save**.

Links between segmentation rules and correlation rules are disabled for the selected tenant.

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 **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.

If the correlator is disabled while processing context table data, the "Internal communication failed" error is displayed in the interface, indicating that the connection to the data source, namely the correlator, has been lost.

When records are deleted from context tables after their lifetime expires, 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 process events and 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

You can view the full list of context tables or the list of context tables linked to a specific correlator.

Full list of context tables

To view the full list of context tables:

In the KUMA web interface, select **Resources** → **Resources** → **Context tables**.

The **Context tables** list is displayed.

List of context tables of a specific correlator

To view the context table list of the correlator:

1. In the KUMA web interface, select **Resources** → **Work with services** → **Active services**.
2. Select the check box next to the correlator for which you want to view the list of context tables.
3. In the toolbar or in the menu (⋮), click **Go to context tables**.

The **Correlator context tables** list is displayed.

The table contains the following data:

- **Name**—name of the context table.
- **Records**—current number of records in 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 KUMA web interface, select the **Resources** section.
2. In the **Resources** section, click **Context tables**.
3. In the **Context tables** window, click **Create new**.

   This opens the **Create context table** window.
4. In the **Name** field, enter a name for the context table.
5. In the **Tenant** drop-down list, select the tenant that owns the resource.
6. 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.
7. In the **Tags** drop-down list, select the tags for the context table 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.
8. In the **Description** field, provide any additional information.

   You can use up to 4,000 Unicode characters.

   This field is optional.
9. On the **Schema** tab, 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 of the table must be a key field. All field names 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	—
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).

10. Add the required number of context table rows.

> After the context table is saved, changing the number of rows is not possible.

11. Click the **Create** button.

The context table is added.

After adding the context table, link it to a correlation rule in the correlation rule settings in the **Context tables update** section. Then link this correlation rule to a correlator.

Once you linked the correlation rule to a correlator, the **Correlator** field and the **Content** section appear in the context table settings. From this section you can manage the context table contents.

Viewing context table settings

To view the context table settings:

1. In the KUMA web interface, select the **Resources** section.
2. In the **Resources** section, click **Context tables**.
3. 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 KUMA web interface, select the **Resources** section.
2. In the **Resources** section, click **Context tables**.
3. In the list in the **Context tables** window, select the context table whose settings you want to edit.
4. Specify the values of the following parameters:
   - **Name**—unique name of the resource.
   - **Tenant**—the tenant that owns the resource. This field is not editable.
   - **TTL**—the record added to the context table is stored in it for this duration. This value is specified in seconds.
   - **Tags** for resource search.
   - **Description**—any additional information about the resource.
   - **Correlator**—the correlator to which the correlation rule that uses the context table is linked. This field is not editable. This field appears after you link the correlation rule to a correlator.
   - **Content** is the content of the context table. This section appears after you link the correlation rule to a correlator.

     How to change the content of the context table

To manually populate the context table or edit its content, you can:

- Add, edit, or delete a row in the table
- Duplicate a row

  Instead of creating a new row, you can select a row and duplicate it, edit the field values, and save.
- Export the context table
- Import the context table

5. Go to the **Schema** tab. This tab displays 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.
6. 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. To apply the changes (delete the table), update the configuration of the correlator: in the **Resources** → **Active services** section, in the list of services, select the check box next to the relevant correlator and click **Update configuration**.
7. Add the context table in which you edited the settings to the correlator.
8. To apply the changes (add a table), update the configuration of the correlator: in the **Resources** → **Active services** section, in the list of services, select the check box next to the relevant correlator and click **Update configuration**.
9. 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 7.
10. Import the adapted data to the context table.

The configuration of the context table is updated.

Duplicating context table settings

To copy a context table:

1. In the KUMA web interface, select the **Resources** section.
2. In the **Resources** section, click **Context tables**.
3. Select the check box next to the context table that you want to copy.
4. Click **Duplicate**.
5. Specify the necessary settings.
6. 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 KUMA web interface, select the **Resources** section.
2. In the **Resources** section, click **Context tables**.
3. 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.
4. Click the **Delete** button.
5. Click **OK**.

The context tables are deleted.

Viewing and finding records in the context table

To view a list of context table records:

1. Go to the full list of context tables or to the list of context tables of a specific correlator.
2. In the **Name** column, select the relevant context table.

The list of records for the selected context table is displayed.

To search the table, in the **Search** field, enter the value of a record key or the value of a field. The list of context table records displays only the records that contain 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.

The list of records 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. The date and time format depends on the selected solution 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`

  If the **Expires** 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**—date and time of the last update of the context table.

Adding, editing, and deleting records in the context table

To edit context table content:

1. Open the context table that you want to edit.

2. Do one of the following:

   - To add a row, click the **Add** button, then specify the field values and click the checkmark icon (✓) to save your changes. The fields indicated on the **Schema** tab as key fields must be filled.

     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 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	[]

- To edit a row, in the **Actions** column, click the pencil icon ( ), then specify the field values and click the checkmark icon ( ) to save your changes.

  The value of the key field of the record is not available for editing. You can change it by exporting and importing a record.

  The values of the **Timestamp** and **Timestamp list** fields must be greater than or equal to 0. The values of the **IP address** and **IP address list** fields must conform to the IPv4 or IPv6 format.

- To duplicate a row, select a row and click the **Duplicate** button, then specify the field values and click the checkmark icon ( ) to save your changes.

- To delete a row, in the **Actions** column, click the trash can icon ( ).

The context table is modified.

Exporting data from a context table

To export data from a context table:

1. In the KUMA web interface, select the **Resources** → **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 **Timestamp** type field depends on the selected solution 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 the TSV format, the tabulation character.

You can also export data into the context table in the context table settings window.

Importing data into a context table

To import data to a context table:

1. In the KUMA web interface, select the **Resources** section.
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.

You can also import data into the context table in the context table settings window.

Example of incident investigation using Kaspersky SIEM

Detecting an attack on the corporate IT infrastructure using Kaspersky SIEM solution includes the following steps:

1. Preliminary steps
2. Assigning an alert to a user
3. Check if the triggered correlation rule matches the data of the alert events
4. Analyzing alert information
5. False positive check
6. Determining alert severity
7. Incident creation
8. Investigation
9. Searching for related assets
10. Searching for related events
11. Recording the causes of the incident
12. Response
13. Restoring assets operability
14. Closing the incident

The description of the steps provides an example of response actions that an analyst might take when an incident is detected in the organization's IT infrastructure. You can view the description and example for each step by clicking the link in its title. The examples are directly relevant to the step being described.

For conditions of the incident for which examples are provided, see the *Incident conditions* section.

For more information about response methods and tools, see the *Incident Response Guide*. On the Securelist website by Kaspersky, you can also find additional recommendations for incident detection and response.

Incident conditions

Parameters of the computer (hereinafter also referred to as "asset") on which the incident occurred:

- Asset operating system – Windows 10.
- Asset software – Kaspersky Endpoint Agent, Kaspersky Endpoint Security.

KUMA settings:

- Integration with Active Directory, Kaspersky Security Center, Kaspersky Endpoint Detection and Response is configured.
- *SOC_package* correlation rules from the solution distribution kit are installed.

A cybercriminal noticed that the administrator's computer was not locked, and performed the following actions on this computer:

1. Uploaded a malicious file from his server.
2. Executed the command for creating a registry key in the `\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` hive.
3. Added the file downloaded at the first step to autorun using the registry.
4. Cleared the Windows Security Event Log.
5. Completed the session.

Step 1. Preliminary steps

Preliminary steps are as follows:

1. Event monitoring.

   When a collector is created and configured in KUMA, the solution writes information security events registered on controlled elements of the organization's IT infrastructure to the event database. You can find and view these events.

2. Creating a correlator and correlation rules.

   When a sequence of events that satisfy the conditions of a correlation rule is detected, the solution generates alerts. If the same correlation rule is triggered for several events, all these events are associated with the same alert. You can use correlation rules from the distribution kit or create them manually.

3. Configuring email notifications about an alert to one or more email addresses.

   If notification is configured, KUMA sends a notification to the specified email addresses when a new alert is received. The alert link is displayed in the notification.

4. Adding assets.

You can only perform response actions for an asset (for example, block a file from running) if the asset is added to KUMA.

Performing response action requires integrating KUMA with Kaspersky Security Center and Kaspersky Endpoint Detection and Response.

Example

The analyst has carried out the following preliminary steps:

- Installed the *SOC_package* correlation rules from the distribution kit and linked them to the correlator.
- Configured the sending of alert notifications to the analyst's email.
- Imported assets from Kaspersky Security Center to KUMA.

According to the incident conditions, after the administrator logged into their account, a malicious file was run, which the attacker had added to Windows autorun. The asset sent Windows security event log events to KUMA. The correlation rules were triggered for these events.

As a result, the following alerts were written to the KUMA alert database:

- R223_Collection of information about processes.
- R050_Windows Event Log was cleared. R295_System manipulations by a non-privileged process.
- R097_Startup script manipulation.
- R093_Modification of critical registry hives.

The information about the alert contains the names of the correlation rules based on which the alerts were created, and the time of the first and last event created when the rules were triggered again.

The analyst received alert notifications by email. The analyst followed the link to the *R093_Changes to critical registry hives* alert from the notification.

Step 2. Assigning an alert to a user

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

Example

As part of the incident, the analyst assigns the alert to themselves.

Step 3. Check if the triggered correlation rule matches the data of the alert events

At this step, you must view the information about the alert and make sure that the alert event data matches the triggered correlation rule.

Example

The name of the alert indicates that a critical registry hive was modified. The **Related events** section of the alert details displays the table of events related to the alert. The analyst sees that the table contains one event showing the path to the modified registry key, as well as the original and the new value of the key. Therefore, the correlation rule matches the event.

Step 4. Analyzing alert information

At this step, analyze the information about the alert to determine what data is required for further analysis of the alert.

Example
From the alert information, the analyst learns the following:

- Which registry key has been modified
- On which asset
- The name of the account used to modify the key

This information can be viewed in the details of the event that caused the alert (**Alerts** → *R093_Modification of critical registry hives* → **Related events** → event 2022-08-23 17:27:05), in the FileName, DeviceHostName, and SourceUserName fields respectively.

Step 5. False positive check

At this stage, make sure that the activity that triggered the correlation rule is abnormal for the organization IT infrastructure.

Example
At this step, the analyst checks whether the detected activity can be legitimate as part of normal system operation (for example, an update). The event information shows that a registry key was created under the user account using the *reg.exe* utility. A registry key was also created in the `\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` hive, responsible for autorun of applications at user logon. Based on this information, one can surmise that the activity is not legitimate and the alarm is not false.

Step 6. Determining alert severity

You can change the alert severity level, if necessary.

Example
The analyst assigns a high severity to the alert.

Step 7. Incident creation

If steps 3 to 6 reveal that the alert warrants investigation, you can create an incident.

Example
The analyst creates an incident in order to perform an investigation.

Step 8. Investigation

This step includes viewing information about the assets, accounts, and alerts related to the incident in the incident information section.

Information about the impacted assets and accounts is displayed on the **Related assets** and **Related users** tabs in the incident information section.

Example

The analyst opens the information about the affected asset (**Incidents** → the relevant incident → **Related alerts** → the relevant alert → **Related endpoints** → the relevant asset). The asset information shows that the asset belongs to the *Business impact/HIGH* and *Device type/Workstation* categories, which are critical for the organization IT infrastructure.

The asset information also includes the following useful data:

- FQDN, IP address, and MAC address of the asset.
- The time when the asset was created and the information was last updated.
- The number of alerts associated with this asset.
- The categories to which the asset belongs.
- Asset vulnerabilities.
- Information about the installed software.
- Information about the hardware characteristics of the asset.

The analyst opens the information about the associated user account (**Incidents** → the relevant incident → **Related alerts** → link with the relevant alert → **Related users** → account).

The following account information may be useful:

- User name.
- Account name.
- Email address.
- Groups the account belongs to.
- Password expiration date.
- Password creation date.
- Time of the last invalid password entry.

Step 9. Searching for related assets

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

Example

The analyst checks for other alerts that occurred on the assets related to the incident (**Incidents** → the relevant incident → **Related alerts** → the relevant alert → **Related endpoints** → the relevant asset → **Related alerts**). In the alert window, you can configure filtering by time or status to exclude outdated and processed alerts. The time when the asset alerts were registered helps the analyst to determine that these alerts are related, so they can be linked to the incident (select the relevant alerts → **Link** → the relevant incident → **Link**).

The analyst also finds the associated alerts for the account and links them to the incident. All related assets that were mentioned in the new alerts are also scanned.

Step 10. Searching for related events

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

The events can be found in the KUMA event database manually or by selecting any of the related alerts and clicking **Find in events** in the alert details (**Incidents** → the relevant incident → **Related alerts** → the relevant alert → **Related endpoints** → **Find in events**). The found events can be linked to the selected alert, however, the alert must be unlinked from the incident before that.

Example

As a result, the analyst found the *A new process has been created* event, where the command to create a new registry key was recorded. Based on the event data, the analyst detected that `cmd.exe` was the parent process for `reg.exe`. In other words, the cybercriminal started the command line and executed the command in it. The event details include information about the *ChromeUpdate.bat* file that was autorun. To find out the origin of this file, the analyst searched for events in the event database using the `FileName = ‘C:\\Users\\UserName\\Downloads\\ChromeUpdate.bat’` field and the `%%4417` access mask (access type *WriteData (or AddFile)*):

```
SELECT * FROM 'events' WHERE DeviceCustomString1 like '%4417%' and FileName like
‘C:\\Users\\UserName\\Downloads\\ChromeUpdate.bat’ AND Device Vendor 'Microsoft' ORDER BY Timestamp DESC LIMIT 250
```

As a result, the analyst discovered that the file was downloaded from an external source using the `msedge.exe process.` The analyst linked this event to the alert as well.

Search for the related events for each incident alert allows the analyst to identify the entire attack chain.

Step 11. Recording the causes of the incident

You can record the information necessary for the investigation in the incident change log.

Example
Based on the results of the search for incident-related events, the analyst identified the causes of the incident and recorded the results of the analysis in the **Change log** field in incident details to pass the information to other analysts.

Step 12. Incident response

You can perform the following response actions:

1. Isolate the asset from the network.
2. Perform a virus scan.
3. Prevent the file from running on assets.

The listed actions are available if KUMA is integrated with Kaspersky Security Center and Kaspersky Endpoint Detection and Response.

Example
The analyst has information about the incident-related assets and the indicators of compromise. This information helps select the response actions.

As part of the incident being considered, it is recommended to perform the following actions:

- Start an unscheduled virus scan of the asset where the file was added to autorun.
  The virus scan task is started by means of Kaspersky Security Center.
- Isolate the asset from the network for the period of the virus scan.
  The asset isolation is performed by means of Kaspersky Endpoint Detection and Response.
- Quarantine the *ChromeUpdate.bat* file and create the execution prevention rules for this file on other assets in the organization.
  An execution prevention rule for a file is created by means of Kaspersky Endpoint Detection and Response.

Step 13. Restoring assets operability

After the IT infrastructure is cleaned from the malicious presence, you can disable the prevention rules and asset network isolation rules in Kaspersky Endpoint Detection and Response.

Example
After the investigation, response, and cleanup of the organization IT infrastructure from the traces of the attack, restoration of the asset operation can be started. For this purpose, the execution prevention rules and the network asset isolation rules can be disabled in Kaspersky Endpoint Detection and Response if they were not disabled automatically.

Step 14. 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.

Analytics

KUMA provides extensive analytics on the data available to the solution from the following sources:

- Events in storage
- Alerts
- Assets
- Accounts imported from Active Directory
- Data from collectors on the number of processed events
- Metrics

You can configure and receive analytics in the **Dashboard**, **Reports**, and **Source status** sections of the KUMA web interface. Analytics are built by using only the data from tenants that the user can access.

The date format depends on the localization language selected in the settings of the solution. Possible date format options:

- English localization: YYYY-MM-DD.
- Russian localization: DD.MM.YYYY.

Working with events

In the **Events** section of the KUMA web interface, you can inspect events received by the solution to investigate security threats or create correlation rules. The events table displays the data received after the SQL query is executed.

Events can be sent to the correlator for a retroscan.

The event date format depends on the localization language selected in the solution settings. Possible date format options:

- English localization: YYYY-MM-DD.
- Russian localization: DD.MM.YYYY.



Filtering and searching events

The **Events** section of the KUMA web interface does not show any data by default. To view events, you need to define an SQL query in the search field and click the **Run query** button. The SQL query can be entered manually or it can be generated using a query builder.

Data aggregation and grouping is supported in SQL queries.

You can search for events across multiple storages. For example, you can find events to determine where a user account is being blocked or which IP addresses were used to log in to which URLs. Example query for finding a blocked user account:

```
SELECT * FROM `events` WHERE DestinationUserName = 'username' AND DeviceEventClassID =
'4625' LIMIT 250
```

To search events across multiple repositories, in the **Events** section, in the drop-down list at the top right of the section, select the check boxes next to the repositories you want.

The list displays the following storages:

- Storages of the Main tenant.
- Available storages of tenants that satisfy one of the following conditions:
  - The tenant that owns the storage is enabled in the tenant filter and the user has permissions to read events in this tenant.
  - The user has access to the tenant of one of the partitions of the storage and has permissions to read events in this tenant.

    For example, if you have access to the collector tenant, but do not have access to the storage tenant, by default, the inaccessible tenant's storage is not displayed in the list of available storages. If a destination in the storage of a tenant that is not available to you is added to the collector of an available tenant, after an event arrives in the partition of the tenant of the collector, the storage of the unavailable tenant appears in the list of storages in the **Events** section.

The drop-down list of storages in the upper right part of the **Events** section displays the name of the first of the selected storages and the number of selected storages, if there are several. You can hover over the drop-down list to display all of the selected storages. If a space in a storage is deleted, the storage drop-down list shows "deleted<ID of deleted space>", and the events remain searchable for the duration of the TTL.

The tenants selected in the tenant filter affect which storages are displayed in the drop-down list of storages. If you disable tenants whose storages are available to you in the tenant filter, these storages are no longer displayed in the drop-down list of storages. If these storages had been selected in the drop-down list of storages, their check boxes are cleared and events from these storages are not displayed. If only one storage is selected in the drop-down list of storages that is not from the Main tenant, and if in tenant selection you disabled the tenant that owns the selected storage, this storage is not displayed in the list of storages and KUMA automatically changes the selection to one of the storages of the Main tenant.

A simple query to all selected storages is allowed, as in the example above. If at least one of the selected storages is not available for the query, KUMA returns an error.

Limitations for searching events across multiple storages:

- When querying multiple storages, export to TSV, retroscan, or REST API requests are not available.
- A SELECT can contain only * and/or names of event fields. Aliases, functions, expressions are not allowed.
- An ORDER BY clause must also contain only event fields (no functions, constants, expressions, and so on). If a field is not present in the list of fields for the SELECT, such a field is automatically added when sending to a specific cluster. You cannot set an ORDER BY ClusterID.
- GROUP BY is not available.

Complex queries with grouping and aggregation are allowed for a single selected storage.

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:

- **Changing a query from the Statistics window**

  *To change the filtering settings in the **Statistics** window:*

  1. Open **Statistics** details area by using one of the following methods:

     - In the ••• drop-down list in the top right corner of the events table select **Statistics**.
     - In the events table click any value and in the opened context menu select **Statistics**.

     The **Statistics** details area appears in the right part of the web interface window.

  2. Open the drop-down list of the relevant parameter and hover your mouse cursor over the necessary value.
  3. Use the plus and minus signs to change the filter settings by doing one of the following:

     - If you want the events selection to include only events with the selected value, click the + icon.
     - If you want the events selection to exclude all events with the selected value, click the − icon.

     As a result, the filter settings and the events table will be updated, and the new search query will be displayed in the upper part of the screen.

- **Changing a query from the events table**

  To change the filtering settings in the events table:

  1. In the **Events** section of the KUMA web interface, click any event parameter value in the events table.
  2. In the opened menu, select one of the following options:

     - If you want the table to show only events with the selected value, select **Filter by this value**.
     - If you want to exclude all events with the selected value from the table, select **Exclude from filter**.

     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.

- **Changing a query from the Event details area**

  To change the filter settings in the event details area:

  1. In the KUMA web interface, in the **Events** section, click the relevant event.

     The **Event details** area appears in the right part of the window.

  2. Change the filtering settings by clicking the plus or minus icons:

     - If you want to apply the filter instantly and update the events table, click + or − next to the parameters that you need.
     - If you want to continue to change filtering parameters, hold down the **Ctrl** key and click + or −. This adds the condition to the body of the query, but does not automatically run the query.

When adding a condition to the query, an expression for the selected parameter is added to the `WHERE` clause. When adding conditions for multiple parameters, the expressions are joined with the `AND` operator. A field to be filtered by is added to the `SELECT` clause only if the `*` character is not used in the query.

Example:
Original query:
```
SELECT *
FROM `events`
ORDER BY Timestamp DESC
LIMIT 250
```
All events are selected without filters, the last 250 events are displayed in descending order by the `Timestamp` field.
When adding new conditions by **`Ctrl`**-clicking the + icon, the query looks like this:
```
SELECT *
FROM `events`
WHERE DeviceProduct = 'Windows' AND StartTime = 1755767943436
ORDER BY Timestamp DESC
LIMIT 250
```
Conditions are added to the `WHERE` clause with the `AND` operator. Fields are not added to the `SELECT` clause because the `*` character is used in this clause. The query is not executed automatically.

3. After adding all the necessary filtering conditions, run the query in one of the following ways:

   - Add the last condition without holding down the **`Ctrl`** key. The query is run automatically.
   - Close the event details area and run the query manually by clicking **Run query** button or pressing **`Ctrl+Enter`**.

   The filter settings and the events table are updated, and the modified search query is displayed in the upper part of the screen.

After modifying a query, all query parameters, including the added filter conditions, are transferred to the query builder and the search field.

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. Also, the query created in the builder does not overwrite the query that was entered into the search string until you click the **Apply query** button in the builder window.

In the SQL query input field, you can enable the display of control characters.

You can also filter events by time period. Search results can be automatically updated.

The filter configuration can be saved. Existing filter configurations can be deleted.

Filter functions are available for users regardless of their roles.

When accessing certain event fields with IDs, KUMA returns the corresponding names.

For more details on SQL, refer to the ClickHouse documentation. For SQL operators and functions supported in KUMA, see also the KUMA operator usage and supported functions.

Selecting Storage

Events that are displayed in the **Events** section of the KUMA web interface are retrieved from storage (from the ClickHouse cluster). Depending on the demands of your company, you may have more than one Storage. However, you can only receive events from one Storage at a time, so you must specify which one you want to use.

To select the Storage you want to receive events from,

In the **Events** section of the KUMA web interface, open the drop-down list of storages in the upper right part of the section and select one or more spaces of one or more storages.

The list displays the following spaces:

- Spaces added when the storage service was created.
- The KUMA Default space.
- The KUMA Audit space, if the Main tenant is selected and if there is at least one audit event in this space.

KUMA displays only those spaces to which the user has access permissions.

Events from the selected storage are displayed in the table of events. Names of the selected storages are displayed in the drop-down list.

If you have access rights to the space that stores events of a tenant to which you do not have access permissions, such a space is displayed in the list, but is not searchable.

Generating an SQL query using a builder

In KUMA, you can use a query builder to generate an SQL query for filtering events.

To generate an SQL query using a builder:

1. In the **Events** section of the KUMA web interface, click the button.

   The filter constructor window opens.

2. 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 make viewing the search results easier, select the necessary fields in the drop-down list. In this case, the data only for the selected fields is displayed in the table. Note that Select * increases the duration of the request execution, but eliminates the need to manually indicate 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**.

   If you use aggregation functions in an SQL query, you can customize the display of columns in the event table. After executing the query without *, the columns will match the query.

If the query uses grouping or aggregation functions, instead of one number that was received as a response to the `SELECT max(BytesOut) FROM `events` LIMIT 250` query, the radar shows a distribution of events from which this number is obtained.

When filtering by alert-related events in alert investigation mode, you cannot perform operations on the data of event fields or assign names to the columns of displayed data.

- **FROM**—data source. Select the **events** value.
- **WHERE**—conditions for filtering events.

  Conditions and groups of conditions can be added by using the **Add condition** and **Add group** buttons. The **AND** operator value is selected by default in the group of conditions, but you can change the operator by clicking it. Available values: **AND**, **OR**, **NOT**. The structure of conditions and condition groups can be changed by using the ⠿ icon to drag and drop expressions.

  Adding 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.

  You can delete filter conditions and filter condition groups by clicking ×.
- **GROUP BY**—event fields or aliases to be used for grouping the returned data.

  If you use data grouping in an SQL query, custom fields specified in the query are displayed in the list of events. Table columns in a group cannot be customized. In group events, you can customize the display of columns.

  When filtering by alert-related events in alert investigation mode, you cannot group the returned data.
- **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 user-defined period and the number of strings in the search results exceeds the defined value, you can click the **Show next** button to display additional strings in the table. This button is not displayed when filtering events by the standard period.

3. Click **Apply query**.

   The current SQL query will be overwritten. The generated SQL query is displayed in the search field.

   If you want to reset the builder settings, click the **Default query** button.

   If you want to close the builder without overwriting the existing query, click the button × in the upper right part of the query creation window.

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 background** button. In this case, the following happens:

- The query is queued for execution in the background, and a task is created in the **Task Manager** 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 **Task manager** section.

If necessary, you can cancel the query. To do this, click the arrow icon next to the **Run query** button and select **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.

For more details on SQL, refer to the ClickHouse documentation. See also KUMA operator usage and supported functions.

Manually creating an SQL query

You can use the search field manually to create SQL queries of any complexity to filter events.

To manually generate an SQL query:

1. Go to the **Events** section of the KUMA web interface.

   An input form opens.

2. Enter your SQL query into the input field. You must use single quotes in your queries.

3. Click **Run query** to run the query.

   A table of events that satisfy the criteria of your query will be displayed. If necessary, you can filter events by period.

Supported functions and operators

Function	Description
`SELECT`	Event fields that you want to be returned. The following functions and operators are supported: • Aggregation functions: `count`, `avg`, `max`, `min`, `sum`. • Arithmetic operators: `+`, `-`, `*`, `/`, `<`, `>`, `=`, `!=`, `>=`, `<=`. You can combine functions and operators in an SQL query. If you use aggregation functions in an SQL query, you can customize the display of columns in the event table. After executing the query without *, the columns will match the query. If the query uses grouping or aggregation functions, instead of one number that was received as a response to the `SELECT max(BytesOut) FROM `events` LIMIT 250` query, the radar shows a distribution of events from which this number is obtained.

Function	Description
`FROM`	Data source. When creating an SQL query, you need to specify the events value as the 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 SQL queries) You must additionally escape special characters with the backslash `\`.
`GROUP BY`	Event fields or aliases to be used for grouping the returned data. If you use data grouping in an SQL query, custom fields specified in the query are displayed in the list of events. Table columns in a group cannot be customized. In group events, you can customize the display of columns.
`ORDER BY`	Columns by which you want to sort the returned data. Possible values: • `DESC` to sort in descending order • `ASC` to sort in ascending order
`OFFSET`	The number of rows to skip before displaying the results of the SQL query.
`LIMIT`	The number of rows that can be displayed in the table. The default value is `250`. If you are filtering events by user-defined period and the number of rows in the search results exceeds the defined value, you can click the **Show next** button to display additional rows in the table. This button is not displayed when filtering events by the standard period.

Examples of SQL 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.

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.

Example:

```
SELECT * FROM `events` WHERE match(Message, 'ssh:\'connection.*') ORDER BY Timestamp DESC LIMIT 250
```

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.

When creating a normalizer for events, you can choose whether to retain the field values of the raw event. The data is stored in the **Extra** event field. This field is searched for events by using the LIKE operator.

Example:

```
SELECT * FROM `events` WHERE DeviceAddress = '00.00.00.000' AND Extra LIKE '%"app":"example"%' ORDER BY Timestamp DESC LIMIT 250
```

In the events table, all events for hosts with the IP address 00.00.00.000 where the example process is running are sorted by the **Timestamp** column in descending order. The number of strings that can be displayed in the table is 250.

If you created an SQL query manually in the search field and then switched to the builder, the SQL query parameters are not transferred to the builder. In this case, you will need to re-create the SQL query in the builder. The SQL query created in the builder does not overwrite the SQL query that was entered into the search string until you click the **Apply query** button in the builder window. If you created an SQL query in the query builder and then switched to the search field, the query parameters are transferred automatically.

Aliases must not contain spaces.

For more details on SQL, refer to the ClickHouse documentation. See also the supported ClickHouse functions.

Saving query history

In the KUMA web interface, the **Events** section keeps a personal history of SQL search queries. You can refer to the history of queries and quickly find a query you have used in the past.

The queries are saved in the browser. Changing your browser resets the query history. The history stores the last 32 of your queries. Queries are displayed with formatting. The date and time when the query was executed is displayed.

Only queries with correct syntax are saved in the query history.

To find a search query in history:

1. In the **Events** section, place the cursor in the query bar.
2. Click the (**SQL history**) button in the toolbar of the query window or press **Ctrl+]** or **Ctrl+[**.

   This opens the query history window. The last query you ran is displayed at the top of the list of queries.
3. Select a search query. To navigate the list of saved queries, press **Ctrl+]** or **Ctrl+[**.

   The selected query is pasted to the search bar.
4. If necessary, run the search query. To do so, press **Ctrll+Enter**.

   When you run a query, the query history window is closed.

You can also close the query history window by pressing the **Esc** key; this moves the cursor to the query editing window.

You can copy the query to the clipboard by pressing **Ctrl+C**. This copies the part of the query that is highlighted with the mouse, or the entire query string, limited by the box, if there was no selection.

You can save the query by pressing **Ctrl+S**, or by clicking the button (**Save query**) in the row. This closes the history window and opens the sidebar for saving the query with populated fields.

You can paste a query into the query window without running it by clicking the (**Insert query**) button, or by pressing **Enter**.

Managing saved search queries

In KUMA, in the **Events** section, you can organize saved SQL queries in a tree of folders for structured storage and quick search of SQL queries. You can edit previously saved queries, rename them, arrange queries in groups (folders) of accessible tenants, and search for previously saved queries in the search bar.

Saved queries follow the tenant access model and are visible to all users that have access to the corresponding tenants.

Saved queries are placed in the **Saved queries pane.** You can open or close the **Saved queries** pane using the (bookmark) button on the toolbar of the query window.

The **Saved queries** navigation pane on the left contains the following:

- The query search window where you can search by folder name and query name.
- Tree of folders and queries, the **Favorites** folder.
- The query filter that displays the queries you have created (on the **My** tab) and all queries (on the **All** tab)

Displaying a query in the query window

To display a query in the query window:

1. In the **Events** section, click the (bookmark) button.

   This opens the **Saved queries** navigation pane.

2. In the tree of folders in the **Saved queries** pane, click to select the relevant query.

The text of the selected query is displayed in the query window.

Saving a query to a folder or a tenant

In the **Events** section, after entering a query in the query window and clicking the **Save current query** button (floppy disk icon), you can save the SQL query to one of the folders or tenants that you can select.

To save a query to a folder:

1. In the **New search** window, in the **Filter name** field, enter a name for the filter.

   Consider the following when naming the filter:

   - The name can contain Unicode characters.
   - The minimum length of a name is 1 character, and the maximum length is 128 characters.
   - Names must be unique within a tenant.
   - Tab characters, new line characters, and paragraph characters are replaced with space characters.

When saving the name:

- Leading and trailing spaces are removed.
- Multiple spaces between characters are replaced with a single space character.

2. In the **Query** field, enter the text of the SQL query.
3. Select a tenant, a folder from the list of folders created in the tenant that is available to you, or create a new folder by clicking the **Add folder** button.

   You can also add a folder in the **Saved queries** window: click the + button next to the tenant to open the **New folder** window and add the folder in that window.
4. Click the **Save** button.

The query is saved in the selected folder or tenant.

Viewing the text of a query

To view the text of a query:

1. In the **Events** section, click the (bookmark) button.

   This opens the **Saved queries** navigation pane.
2. In the tree of folders in the **Saved queries** pane, click the button next to the relevant query.

This opens a window with the text of the query.

Setting the default query

You can set your default query that the **Events** section displays when opened.

To set the default query:

1. In the **Events** section, click the (bookmark) button.

   This opens the **Saved queries** navigation pane.
2. In the tree of folders on the **Saved queries** pane, select the relevant query.
3. Click the button and in the displayed menu, select **Use by default**.

The selected query becomes the default query.

Adding folders and queries to favorites

To add a folder or individual query to favorites:

1. In the **Events** section, click the (bookmark) button.

   This opens the **Saved queries** navigation pane.

2. In the tree of folders in the **Saved queries** pane, select the relevant folder or query.
3. Click the ⋮ button and in the displayed menu, select **To favorites**.

The selected folder or resource is added to the **Favorites** folder.

Editing a query

To edit a query:

1. In the **Events** section, click the (bookmark) button.
   This opens the **Saved queries** navigation pane.
2. In the tree of folders on the **Saved queries** pane, select the relevant query.
3. Click the ⋮ button and in the displayed menu, select **Edit**.
   This opens the window for editing the saved query.
4. In the window, edit the query name or text, or the folder in which you want to save the query.
5. Click the **Save** button.

The query is updated and saved in the selected folder.

Deleting a query

To delete a query from a folder (if you have sufficient rights):

1. In the **Events** section, click the bookmark () icon.
   This opens the **Saved queries** navigation pane.
2. In the tree of folders on the **Saved queries** pane, select the relevant query.
3. Click the three-dot icon (⋮) and in the menu that appears, select **Delete**.
4. Confirm deletion in the displayed window.

The query is deleted.

Filtering events by period

In KUMA, you can specify the time period to display events from.

To filter events by period:

1. In the **Events** section of the KUMA web interface in the upper part of the window, open the time period drop-down list in the top right part of the section.

2. Select a period in one of the following ways:

   - If you want to specify an **exact date**, select start and end dates of the period in the calendar 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 is populated 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. Once 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 periods available in the **Relative period** list.

     The period is calculated relative to the current time.

   - If you want to specify a **custom period**, set or edit the value of the **Range from** and **Range to** fields 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 period relative to the current time as a formula. You can also combine these methods. 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.

The default setting is **5 minutes** (**now-5m**). 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.

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 the **Today** period, events are displayed for the period from 03:00:00.000 until now, not from 00:00:00.000 until now.

If, when selecting a relative period, such as **Today**, **Yesterday**, **This week**, or **This month**, you want to take your time zone into account when displaying events, you need to manually add a time offset in the **Range from** and **Range to** fields (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 of yesterday, you can use an SQL query with a filter by Timestamp or specify an exact date and time.

3. If you want to capture the last relative period for which a query was executed as an exact date and time, click the button.

   The relative period is then converted to an exact date and time. Hovering over this button displays the date and time that will be captured. Relative periods (**Today**, **Yesterday**, **This week**, **This month**) are converted to an exact UTC date and time, other periods are converted to a date and time that takes into account the time zone of the browser.

   If you have executed an SQL query with event grouping, when switching between groups in the query results, the relative time range is automatically converted to an exact date and time.

4. Click **Run query** to run the query.

   Only events registered during the specified time period are displayed. The period will be displayed in the upper part of the window.

You can also configure the display of events by using the events histogram that is displayed when you click the button in the upper part of the **Events** section. Events are displayed if you click the relevant data column or select the relevant time period and click the **Show events** button.

Displaying the list of events with a relative and absolute calendar period

The display of the list of events in the table in the **Events** section depends on the selected event filtering period: relative or absolute.

When a relative period is selected, the display is as follows:

- If a certain value is specified for the `LIMIT` variable in the SQL query, the number of displayed events corresponds to this value.

  Example SQL query: `SELECT * FROM `events` ORDER BY Timestamp DESC LIMIT 10`.

- If the `LIMIT` value is not specified in the SQL query, 250 events are displayed by default.

  Example SQL query: `SELECT * FROM `events` ORDER BY Timestamp DESC`.

If a relative period is selected, all events are displayed on the same page.

If an absolute period is selected, the display is as follows:

- If the value of the `LIMIT` variable in the SQL query is less than or equal to 49, and more than 49 events are found, to view additional rows in the table, you need to click the **Show more** button.

  Example SQL query: `SELECT * FROM `events` ORDER BY Timestamp DESC LIMIT 48`.

- If the value of the `LIMIT` variable in the SQL query is greater than 49, and more than 49 events are found, additional events are loaded into the list automatically.

  Example SQL query: `SELECT * FROM `events` ORDER BY Timestamp DESC LIMIT 50`.

Grouping events

After getting a list of events, you often need to split the events into groups to localize an information security event. KUMA can group events in a list by one or more fields.

To group events, you no longer need to manually edit the text of the query; instead, you can click a column heading in the **Events** section and select **Add "Group by" in query** in the context menu. You can select a sequence of multiple fields to group by, and the fields will be automatically added to the query string. Having selected your fields, click **Run query**. As a result, events are grouped by the specified fields. Found groups are displayed in the **Groups** section. They can be displayed as a table and as cards. You can toggle between the display modes.

You can exclude a group from the query:

- In **Cards** mode, click the (—) button.
- In **Tables** mode, right-click the group and in the context menu, select **Exclude group from filter**.

As a result, the query is automatically modified and the group is excluded from the query.

If you want to go back to the original query, click **Return to initial select query**.

You can navigate through the groups and view the contents of each group.

You can do a global search in all groups or a local search in events within a selected group.

You can use more complex grouping by adding one or more fields.

You can remove a group from the grouping and in this way, go back one step.

If the grouping query returns many events, only the first 1000 events are displayed. If the query contains SELECT Count(ID), you can click the link with the total number of events in the query result to view all events. If the request does not contain Count(ID), the number of events in the group is not indicated, but you still can click the link and view the total number of events in the group.

Statistics, retrospective scan by group, and export to TSV are available.

If you want the grouping result to be independent of time (because events arrive continuously), you can set a fixed relative interval and apply it as an absolute interval so that the events of interest do not drop out of the selection. To fix a relative interval, in the **Events** section, in the time interval drop-down list, select **Apply current range**. You can now manage groups within this query.

If you want to arrange the selected events by months, days, minutes, and seconds, you can group events by the Timestamp field. To group events, select a grouping option in the context menu of the Timestamp field in the event table.

If you want to normalize the value of the Timestamp field and display the time values from different sources in the same UTC time scale, select **Convert to UTC timezone** in the context menu of the Timestamp field in the events table.

Displaying names instead of IDs

When accessing certain event fields with IDs, KUMA returns the corresponding names rather than IDs. This helps make the information more readable. For example, if you access the `TenantID` event field (which stores the tenant ID), you get the value of the `TenantName` event field (which stores the tenant name).

When exporting events, values of both fields are written to the file, the ID as well as the name.

The table below lists the fields that are substituted when accessed:

Requested field	Returned field
`TenantID`	`TenantName`
`ServiceID`	`ServiceName`
`DeviceAssetID`	`DeviceAssetName`
`SourceAssetID`	`SourceAssetName`
`DestinationAssetID`	`DestinationAssetName`
`SourceAccountID`	`SourceAccountName`
`DestinationAccountID`	`DestinationAccountName`

Substitution does not occur if an alias is assigned to the field in the SQL query. Examples:

- `SELECT TenantID FROM `events` LIMIT 250` — in the search result, the name of the tenant is displayed in the TenantID field.
- `SELECT TenantID AS Tenant_name FROM `events` LIMIT 250` — in the search result, the tenant ID will be displayed in the Tenant_name field.

Presets

You can use presets to simplify work with queries if you regularly view data for a specific set of event fields. In the line with the SQL query, you can type `Select *` and select a saved preset; in that case, the output is limited only to the fields specified in the preset. This method slows down performance but eliminates the need to write a query manually every time.
Presets are saved on the KUMA Core server and are available to all KUMA users of the specified tenant.

To create a preset:

1. In the **Events** section, click the ⚙ icon.
2. In the window that opens, on the **Event field columns** tab, select the required fields.

   To simplify your search, you can start typing the field name in the **Search** area.
3. To save the selected fields, click **Save current preset**.

   The **New preset** window opens.
4. In that window, specify the **Name** of the preset, and in the drop-down list, select the **Tenant**.
5. Click **Save**.

   The preset is created and saved.

To apply a preset:

1. In the query entry field, enter **Select ***.
2. In the **Events** section of the KUMA web interface, click the icon.
3. In the opened window, use the **Presets** tab to select the relevant preset and click the button.

   The fields from the selected preset are added to the SQL query field, and the columns are added to the table. No changes are made in Builder.
4. Click **Run query** to run the query.

   After the query execution completes, the columns are filled in.

Limiting the complexity of queries in alert investigation mode

When investigating an alert, the complexity of SQL queries for event filtering is limited if **Related to alert** is selected in the drop-down list of event sources. If this is the case, only the functions and operators listed below are available for event filtering.

> If **All events** is selected from the drop-down list of event sources, these limitations do not apply.

- `SELECT`
  - The `*` character is used as a wildcard to represent any number of characters.
- `WHERE`
  - `AND`, `OR`, `NOT`, `=`, `!=`, `>`, `>=`, `<`, `<=`
  - `IN`
  - `BETWEEN`
  - `LIKE`
  - `inSubnet`

  Examples:
  - `WHERE Type IN ('Base', 'Correlated')`
  - `WHERE BytesIn BETWEEN 1000 AND 2000`
  - `WHERE Message LIKE '%ssh:%'`
  - `WHERE inSubnet(DeviceAddress, '10.0.0.1/24')`
- `ORDER BY`

  Sorting can be done by column.

- `OFFSET`

  Skip the indicated number of lines before printing the query results output.

- `LIMIT`

  The default value is 250.

  If you are filtering events by user-defined period and the number of strings in the search results exceeds the defined value, you can click the **Show next** button to display additional strings in the table. This button is not displayed when filtering events by the standard period.

When filtering by alert-related events in alert investigation mode, you cannot perform operations on the data of event fields or assign names to the columns of displayed data.

Saving and selecting events filter configuration

In KUMA, you can save a filter configuration and use it in the future. Other users can also use the saved filters if they have the appropriate access rights. When saving a filter, you are saving the configured settings of all the active filters at the same time, including the time-based filter, query builder, and the events table settings. Search queries are saved on the KUMA Core server and are available to all KUMA users of the selected tenant.

To save the current settings of the filter, query, and period:

1. In the **Events** section of the KUMA web interface, click the icon next to the filter expression.

   This opens the **New filter** window.

2. In the window that opens, enter the name of the filter configuration in the **Name** field. The name can contain up to 128 Unicode characters.

3. Under **Select folder**, select the tenant to which you want the created filter to belong and the folder inside the tenant, if necessary.

   To simplify your search, you can start typing the name in the **Search** area.

4. If necessary, click **Add folder** to create a new folder inside the selected tenant.

5. Click **Save**.

The filter configuration is now saved.

To select a previously saved filter configuration:

In the **Events** section of the KUMA web interface, click the icon next to the filter expression and select the relevant filter.

The selected configuration is active, which means that the search field is displaying the search query, and the upper part of the window is showing the configured settings for the period and frequency of updating the search results. Click the **Run query** button to submit the search query.

To add the filter configuration to favorites, you can hover over the name of the filter configuration, click the icon next to the name of the configuration and click the **To favorites** button.

Deleting event filter configurations

To delete a previously saved filter configuration:

1. In the **Events** section of the KUMA web interface, click the ☆ icon next to the filter search query.
   This opens the **Saved queries** window.
2. Hover over the configuration that you want to delete and click the ⋮ icon next to its name.
3. This opens a menu; in the menu, click the 🗑 icon.

The filter configuration is now deleted for all KUMA users.

Supported ClickHouse functions

The following ClickHouse functions are supported in KUMA:

- Arithmetic functions.
- Arrays.
- Comparison functions.
- Logical functions.
- Type conversion functions.
- Date and time functions.
- String functions.
- String search functions.
- Conditional functions: only the regular 'if' operator; the ternary operator is not supported.
- Mathematical functions.
- Rounding functions.
- Functions for splitting and merging strings and arrays.
- Bit functions.
- Functions for working with UUIDs.
- Functions for working with URLs.
- Functions for working with IP addresses.
- Functions for working with Nullable arguments.
- Functions for working with geographic coordinates.

Functions from other sections are not supported.

For more details on SQL, refer to the ClickHouse documentation.

Viewing event detail areas

To view information about an event:

1. In the solution web interface window, select the **Events** section.
2. Search for events by using the query builder or by entering a query in the search field.

   The event table is displayed.
3. Select the event whose information you want to view.

The event details window opens.

The **Event details** area appears in the right part of the web interface window and contains a list of the event's 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 the setting value.
- Click a file hash in the **FileHash** field ещ open a list in which you can select one of the following actions:
  - Show info from **Threat Lookup**.

    This is available when integrated with **Kaspersky Threat Intelligence Portal**.
  - Add to **Internal TI of CyberTrace**.

    This is available when integrated with **Kaspersky CyberTrace**.
- Open a window containing information about the asset if it is mentioned in the event fields and registered in the solution.
- Click the link containing the collector name in the **Service** field to view the settings of the service that registered the event.

  You can also link an event to an alert if the solution is in the alert investigation mode and open the **Correlation event details** window if the selected event is a correlation event.
- You can view all non-empty event fields by clicking **Show all event fields**. The following conditions must be satisfied for the rule to trigger:
  - The query is executed for a set of fields and is not a SELECT *.
  - Not all non-empty fields are displayed. If all non-empty fields are displayed in the event card, the **View all event fields** button is not displayed.
  - The event card is opened in KUMA in the **Events** section.

  This does not change the original query or rebuild the table of events.

  Example

  Prerequisite:

An event was sent with multiple non-empty fields, such as Message, SourceAddress, DeviceCustomString1.

A query was executed for a set of fields.

```
SELECT Type, Message

FROM `events`

ORDER BY Timestamp

DESC LIMIT 250
```

Clicking an event opens the **Event Information** window with the following information:

- The Type and Message fields from the query.
- The default TenantID and Timestamp fields not specified in the query are displayed.

Non-empty SourceAddress and DeviceCustomString1 fields of the event that are missing from the query but are present in the sent event are not displayed.

If you want to view all non-empty fields, click **View all event fields**.

This displays all non-empty fields of the event: Type, Message, TenantID, Timestamp, SourceAddress, DeviceCustomString1.

- Change the display of the event by clicking the **Format** button. By default, the event is displayed as a one-line string in XML or JSON format. When you click this button, the event is displayed in multi-line format with indentation. Clicking the button again reverts the display to the one-line format. Formatting is applied to all events in the table in the **Events** section.

In the **Event details** area, the name of the described object is shown instead of its ID in the values of the following settings. At the same time, if you change the filtering of events by this setting (for example, by clicking — to exclude events with a certain setting-value combination from search results), the object's ID, and not its name, is added to the SQL query:

- TenantID
- ServiceID
- DeviceAssetID
- SourceAssetID
- DestinationAssetID
- SourceAccountID
- DestinationAccountID

Exporting events

In KUMA, 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 currently 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 KUMA web interface due to the special features of the SQL query.

To export information about events:

1. In the **Events** section of the KUMA web interface, click the **TSV** button in the upper part of the table of events.

   The new export TSV file task is created in the **Task manager** section.

2. Find the task you created in the **Task manager** section.

   When the file is ready to be downloaded, the **Status** column of the task displays the **Completed** status and the icon.

3. 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.

Configuring the table of events

Responses to user SQL queries are presented as a table in the **Events** section. The fields selected in the custom query appear at the end of the table, after the default columns. This table can be updated.

The following columns are displayed in the events table by default:

- **TenantID**.
- **Timestamp**.
- **Name**.
- **DeviceProduct**.
- **DeviceVendor**.
- **DestinationAddress**.
- **DestinationUserName**.

In KUMA, you can customize the displayed set of event fields and their display order. The selected configuration can be saved to a preset.

When using SQL queries with data grouping and aggregation for filtering events, statistics are not available and the order of displayed columns depends on the specific SQL query.

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.

To configure the fields displayed in the events table:

1. Click the icon in the top right corner of the events table.

   A window for selecting the event fields that should be displayed in the events table will be displayed.

2. Select the check boxes opposite the fields that you want to view in the table. You can search for relevant fields by using the **Search** field.

   You can configure the table to display any event field from the KUMA event data model and the extended event schema. The **Timestamp** and **Name** parameters are always displayed in the table. Click the **Default** button to display only default event parameters in the events table.

   When you select a check box, the events table is updated and a new column is added. When a check box is cleared, the column disappears.

   You can also remove columns from the events table by clicking the column title and selecting **Hide column** from the drop-down list.

3. If necessary, change the display order of the columns by dragging the column headers in the event tables.

4. If you want to sort the events by a specific column, click its title and in the drop-down list select one of the available options: **Ascending** or **Descending**.

The selected event fields will be displayed as columns in the table of the **Events** section in the order you specified.

The list of selected fields is stored locally in your browser settings. To make column display settings identical across different browsers or devices, we recommend using presets.

Refreshing events table

You can update the displayed event selection with the most recent entries by refreshing the web browser page. You can also refresh the events table automatically and set the frequency of updates. Automatic refresh is disabled by default.

To enable automatic refresh,

Select a refresh rate from the drop-down list in the upper-right part of the **Events** section:

- **5 seconds**
- **15 seconds**
- **30 seconds**
- **1 minute**
- **5 minutes**
- **15 minutes**

The events table now refreshes automatically.

To disable automatic refresh:

In the drop-down list of refresh rates in the upper-right part of the **Events** section, select **No refresh**.

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:

Select **Statistics** from the ••• drop-down list in the upper-right corner of the events table, or click on any value in the events table 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, you can also see its five most frequently occurring values. You can find relevant parameters by using the **Search fields** field.

In a high availability configuration, for all event fields that contain the FQDN of the Core, the **Statistics** section displays **core** instead of the FQDN.

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.

Viewing correlation event details

You can view the details of a correlation event in the **Correlation event details** window.

To view information about a correlation event:

1. In the **Events** section of the KUMA web interface, click a correlation event.

   You can use filters to find correlation events by assigning the `correlated` value to the `Type` parameter.

   The details area of the selected event will open.

2. Click the **Detailed view** button.

The correlation event window will open. The event name is displayed in the upper left corner of the window.

You can also navigate to the correlation event window from the the **Related events** section of the alert information window. In this case, you can go back to the alert information window by clicking the link in the upper part of the window. Otherwise, KUMA does not display the link to an alert.

The **Correlation event details** section of the correlation event window contains the following data:

- **Correlation event severity**—the severity of the correlation event.
- **Correlation rule**—the name of the correlation rule that triggered the creation of this correlation event. The rule name is represented as a link that can be used to open the settings of this correlation rule.
- **Correlation rule severity**—the severity of the correlation rule that triggered the correlation event.
- **Correlation rule ID**—the identifier of the correlation rule that triggered the creation of this correlation event.
- **Tenant**—the name of the tenant that owns the correlation event.

The **Related events** section of the correlation event window contains the table of events related to the correlation event. These are base events that actually triggered the creation of the correlation event. When an event is selected, the details area opens in the right part of the web interface window.

The **Find in events** link to the right of the section header is used for alert investigation.

The **Related endpoints** section of the correlation event window contains the table of hosts related to the correlation event. This information comes from the base events related to the correlation event. Clicking the name of the asset opens the **Asset details** window.

The **Related users** section of the correlation event window contains the table of users related to the correlation event. This information comes from the base events related to the correlation event.

Generating an SQL query using KUMA SQL functions

KUMA's SQL functions allow using the attributes of assets and accounts in search queries to filter events, generate reports and widgets (graph type: **Table**). You can enrich events with data from dictionaries, tables, assets, and accounts using the following sets of functions:

- The enrich function set. Allows enriching query results with fields of the asset or account, values from a dictionary of the **Dictionary** type or a dictionary of the **Table** type. The `enrich` function set includes the following functions:
  - `enrich_assets`
  - `enrich_accounts`
  - `enrich_table`
  - `enrich_dictionary`
- The lookup function set. Allows adding conditions for assets and accounts to the search query. The `lookup` function set includes the following functions:
  - `lookup_assets`
  - `lookup_accounts`
  - `lookup_assets_category`

The maximum number of events per query is 10,000.

Using the 'enrich' function set in SQL queries

The `enrich` function set (see the table below) is used in the `SELECT` clause of an SQL query to enrich the results.

When using this set of functions, you cannot group events by enriched fields.

Widgets and reports enriched by the functions must be owned by no more than one tenant.

The values of asset fields that are specified in SQL queries are case-sensitive.

Description of the 'enrich' set of functions

Function	Event field	Asset field	Description
enrich_assets	DeviceAssetID DestinationAssetID SourceAssetID	name os osBuild ipAddress macAddress fqdn source software cii ramSize kscLastAvBasesUpdate kscLastInfoUpdate kscLastUpdate kscLastSystemStart kscStatusMask kscRtProtectionState kscEncryptionState kscAntiSpamStatus kscEmailAvStatus kscDlpStatus kscStatusID kscEdrStatus lastVisible aistatus aiscore	Enriches the query result with asset fields based on the ID specified passed to the function. For example: `enrich_assets(DeviceAssetID, name)`
enrich_accounts	SourceAccountID DestinationAccountID	name guid domain cn dn employee_id email mail_nickname mobile sid sam_name telephone_number principal_name sn type title division department manager location company street_address physical_delivery_office_name uac	Enriches the query result with account fields based on the ID specified passed to the function. For example: `enrich_account(SourceAccountID, name)` This function is available only for tenants that have LDAP enrichment enabled.

Function	Event field	Asset field	Description

`enrich_table`	`Any`	–	Enriches the query result with values from a dictionary of the **Table** type based on event fields in the response from the ClickHouse cluster.
			For example:
			`enrich_table('Main/some-table', [DeviceProduct], 'value')`
			Only one column of the table can be the key, but the record key can be a compound key.
			For example:
			`enrich_table('Main/Dictionary', [BytesOut|BytesIn], 'value2')`
			The `enrich_table` function returns only one column of the table. If you need to call more columns, call the function again specifying the relevant fields.

`enrich_dictionary`	`Any`	–	Enriches the query result with values from a dictionary of the **Dictionary** type based on event fields in the response from the ClickHouse cluster. For example: • `enrich_dictionary('Main/some-dict', [SourceAddress]` • `enrich_dictionary('Main/some-dict', [SourceAddress, DeviceAddress])` You can view the path to the dictionary in the **Resources** → **Dictionaries** section.

Example SQL queries with an enrich function

You can specify an `enrich` function when manually creating SQL queries. Example queries with an `enrich` function

- All events of the 'events' table with the `Name` and `Message` fields, as well as the `macAddress` and `osBuild` asset fields to enrich the result with asset data. The number of rows that can be displayed in the table is 250.
```
SELECT Name,
Message,
enrich_assets(DeviceAssetID, macAddress),
enrich_assets(DeviceAssetID, osBuild)
FROM `events`
WHERE DeviceAssetID != ''
LIMIT 250
```
- All events of the 'events' table with event fields `SourceUserName` and `SourceAddress`, as well as values of the `mobile` field to enrich the result with account data. The number of rows that can be displayed in the table is 10.
```
SELECT SourceUserName,
SourceAddress,
enrich_accounts(SourceAccountID, mobile) as mobile
FROM `events`
WHERE SourceAccountID != ''
LIMIT 10
```
- All events in the 'events' table with `DeviceProduct` event fields with values from the **Main/[OOTB] Linux Auditd record types**. Sorted by **Timestamp** in descending order. The number of rows that can be displayed in the table is 1000.
```
SELECT DeviceProduct,
enrich_table('Main/[OOTB] Linux. Auditd record types', [DeviceProduct], 'Description') as HTTP
FROM `events` WHERE Type!=4
LIMIT 1000
```
- All events in the events table with `DeviceAction` event fields with values from the **Main/[OOTB] KEDR** dictionary. The number of rows that can be displayed in the table is 250.
```
SELECT DeviceAction,
enrich_dictionary('Main/[OOTB] KEDR. AccountType', [BytesOut]) as accountType
FROM `events` WHERE Type!=4
LIMIT 255
```

Using the 'lookup' function set in SQL queries

The `lookup` function set (see the table below) is used in the WHERE clause of an SQL query to add conditions for assets and accounts to the search query.

In queries containing a `lookup` function, KUMA automatically adds the `*ID` field to the `SELECT` clause, which field is used in the `lookup` function in the query. For example, in a query containing the `lookup_assets(DeviceAssetID, osBuild, '=', '19045.000000')` function a `DeviceAssetID` field is added to the `SELECT` clause.

Description of the 'lookup' set of functions

Function type	Event field	Asset field	Operator	Description
lookup_assets	DeviceAssetID SourceAssetID DestinationAssetID	name os osBuild source cii ramSize kscRtProtectionState kscEncryptionState kscAntiSpamStatus kscEmailAvStatus kscDlpStatus kscStatusID kscEdrStatus aistatus aiscore	=	Allows you to add conditions t result for assets whose field v value specified in the query. For values that are number, yo string with the required numbe For example: lookup_assets(DeviceAss '=', '19045.000000') lookup_assets(DeviceAss '=', '3241234234')
		kscLastAvBasesUpdate kscLastInfoUpdate kscLastUpdate kscLastSystemStart lastVisible ramSize aiscore	< > <= >=	Allows you to add conditions f field values that are: • Less than the value speci • Greater than the value sp query • Less than or equal to the the query • Greater than or equal to t specified in the query For values that are number, yo string with the required numbe For example: lookup_assets(DeviceAss '>', '50') Date values can be specified i formats: • Number of seconds: lookup_assets(Device lastVisible, '>', '1 • Number of milliseconds: lookup_assets(Device lastVisible, '<', '1725971837834') • Number of days: lookup_assets(Device lastVisible, '<', 'n
		name os osBuild fqdn software	like	Allows you to add conditions t result for assets whose field v 'like' expression specified in th For more information about th please refer to the SQL Langu Expressions reference. lookup_assets(Destinati name, 'like','test.compa
		ipAddress macAddress fqdn software kscStatusMask	in	Allows you to add to the result whose values match the value query. For example: lookup_assets(DeviceAss 'in', ['test_fqdn1.compa 'test_fqdn2.company.ru'

Function type	Event field	Asset field	Operator	Description
		`ipAddress`	`insubnet`	Allows you to add conditions f addresses in the specified sub address is specified in CIDR fo For example: `lookup_assets(DeviceAss ipAddress, 'insubnet',`
		`ipAddress`	`inrange`	Allows you to add conditions f addresses in the specified ran The first IP address specified i start IP address, and the seco address. For example: `lookup_assets(DeviceAss ipAddress, 'inrange', [ '192.168.1.184'])` The query filters assets with IP range from 192.168.1.1 to 192.168

Function type	Event field	Asset field	Operator	Description
lookup_accounts	SourceAccountID DestinationAccountID	name guid domain cn dn employee_id email mail_nickname mobile sid sam_name telephone_number principal_name sn title division department manager location company street_address physical_delivery_office_name uac	=	Allows you to add conditions t result for accounts whose fiel the value specified in the quer For values that are number, yo string with the required numbe For example: lookup_accounts(SourceA employee_id, '=', '1545
		name guid domain cn dn employee_id email mail_nickname mobile sid sam_name telephone_number principal_name sn type title division department manager location company street_address physical_delivery_office_name	like	Allows you to add conditions t result for accounts whose fiel the 'like' expression specified i For more information about th please refer to the SQL Langu Expressions reference. lookup_accounts(Destina name, 'like', '_ccount1

Function type	Event field	Asset field	Operator	Description

`lookup_assets_category`	`DeviceAssetID`
	`SourceAssetID`
	`DestinationAssetID`
		–	–	Finds all events that have assc the specified folder.
				For example:
				`lookup_assets_category( ['Main/Business Impact']`
				where `Main` is the name of th `Business Impact` is the nar

Example SQL queries with a lookup function

You can specify a `lookup` function when manually creating SQL queries. Example queries with a `lookup` function

- All events of the 'events' table with the `TenantID` and `DeviceProduct` event fields, as well as the `fqdn` asset field, whose value contains the win10 substring. The number of rows that can be displayed in the table is 250.

```
SELECT TenantID, DeviceProduct
FROM `events`
WHERE DeviceAssetID != ''
AND lookup_assets(DeviceAssetID, fqdn, 'match', 'win10')
LIMIT 250
```

- All events in the 'events' table, with the `TenantID` and `DeviceProduct` event fields, and with values of `the EmployeeID` field, whose value is `15452.` The number of rows that can be displayed in the table is 10.

```
SELECT TenantID, DeviceProduct
FROM `events`
WHERE SourceAccountID != ''
AND lookup_accounts(SourceAccountID, employee_id, '=', '15452')
LIMIT 10
```

- All events of the 'events' table with the `TenantID` and `DeviceProduct` event fields, with the folder in which the asset is located. The number of rows that can be displayed in the table is 100.

```
SELECT TenantID, DeviceProduct
FROM `events`
WHERE DeviceAssetID != ''
AND lookup_assets_category(DeviceAssetID, ['Main/Business Impact'])
LIMIT 100
```

Event route tracing

To find out which connection the events were received from, you can use the **Track event route** setting. You may need to trace the route of an event when multiple agents send events to the same collector, for example, in a large infrastructure where agents are installed on dedicated WEC servers and multiple agents send events to the same collector. The event route information can be useful for troubleshooting event routing.

For the **Track event route** toggle switch to become available, you must specify at least one **internal** destination in the agent. Also, to trace event routes, a connector of the **internal** type must be specified in the collector that is receiving events from the agent. After configuring and saving the agent, information about the agent's route is added to the **S.KL_EventRoute** field of the extended event schema.

The **S.KL_EventRoute** field is displayed only for new events that are received by the collector after enabling the **Track event route** setting. Other services through which the event passes, including the collector, correlator (only correlation rules of the 'simple' type), and router, parse the **S.KL_EventRoute** field, and, if the field is not empty, add their own information to the field when they process the event.

If the event passes through multiple hosts on its way from the sender to the agent, the address of the last host that established a connection with the agent is saved in the **S.KL_EventRoute** field. If there is a proxy server between the sender and the agent, the address of this proxy server is written to the field, and no separate proxy server configuration is required on the agent. The tcp/udp/http connectors pass the address of the host from which the event was received via the **internal** protocol to **S.KL_EventRoute**, and the WEC, WMI, and ETW agents specify the host name of the Windows server on which they are installed.

You can enable event route tracing in one of the following ways:

- Create a new agent, specify the **internal** destination in it and enable the **Track event route** option.
- In an existing agent, add a tab with a new connection and specify the **internal** destination on this tab. After you specify the **internal** destination, the **Track event route** option becomes available. Enable the **Track event route** toggle switch. After you save the agent settings, restart the agent to apply the changes. In this case, route tracing only affects the tab on which the **internal** destination is selected, not the entire agent.
- Unlink the automatically created agent from the collector to make the agent available for editing its settings, or duplicate the automatically created agent and proceed to configure the settings in the duplicate agent. After you specify an **internal** destination in an editable agent, the **Track event route** option becomes available. Enable the **Track event route** toggle switch. After you save the agent settings, restart the agent to apply the changes. In this case, route tracing only affects the tabs on which the **internal** destination is selected, not the entire agent.

The **Event tracking log** section appears in the event card, alert card and correlation event card. This section displays information from the **S.KL_EventRoute** field in a processed form. Service IDs are converted to service names and displayed as clickable links. Clicking a service name opens a new browser tab with the card of the service. If you rename the service, the name of the service also changes when the event is displayed in the card, both for new events and for events that have already been received and processed. If you delete a service in the **Active services** section, the **Event tracking log** section displays **Deleted** instead of the hyperlink. The rest of the route information is not deleted and is still displayed, including the connector type, FQDN, the value of the **SourceAddress** field, and other route parameters from the **S.KL_EventRoute** field. To view raw route information, you can add the **S.KL_EventRoute** column to the event table.

Using route information

You can use the 'Extract from JSON' function or other ClickHouse functions to get the parts of the route that you need for debugging. For more information about functions, please refer to the ClickHouse documentation: https://clickhouse.com/docs/en/sql-reference/functions/json-functions#jsonextractstringjson-indices-or-keys

The following table gives examples of queries.

Query example	Description
`SELECT *, simpleJSONExtractRaw(S.KL_EventRoute, 'version') as version FROM `events` Where Type !=4 ORDER BY Timestamp DESC LIMIT 25`	This query displays events for which the version of the collector is displayed in the **version** column.
`SELECT *, simpleJSONExtractRaw(S.KL_EventRoute, 'connectorKind') as connectorKind FROM `events` Where Type !=4 ORDER BY Timestamp DESC LIMIT 250`	This query displays events for which the type of the collector is displayed in the **connectorKind** column.
`SELECT *, simpleJSONExtractRaw(S.KL_EventRoute, 'connectorKind') as connectorKind FROM `events` Where Type !=4 and simpleJSONExtractRaw(S.KL_EventRoute, 'connectorKind') != '"wec"' ORDER BY Timestamp DESC LIMIT 250`	This query displays events received from connectors of a type other than 'wec'.

If want to select by a different value from the **S.KL_EventRoute** field, you can look up the name of your parameter in the events table in the **Raw** column and use it in your query.

Categorization of events

Categorization allows you to use special tags (categories) to describe events of the same type coming from different sources. This helps to find events related to, for example, user authentication or command execution, and helps in writing correlation logic or displaying data in a dashboard or in reports. You can use the KUMA resource set to categorize certain types of events for specific types of event sources.

The following types of KUMA resources are used to categorize events:

- A dictionary with categories of events
- Enrichment rules for enriching events with supplemental information.
- A normalizer for creating an extended event schema field used in enrichment rules.

As part of categorization, the following attributes can be added to an event:

- Object
- Action
- Result
- Threat
- Source type

Additional attributes can be assigned to the event, which are saved in the SA.KL_EventCategory extended event schema field.

Descriptions of the possible values of the attributes used for categorizing events are provided in tables: *Object attribute, Action attribute, Result attribute, Threat attribute, Source type attribute*.

To customize the categorization of events on a collector:

1. Import the [OOTB] Event Categorization package from the KUMA repository.
2. In the collector, at the **Event enrichment** step, apply an event enrichment rule.

   You can apply the enrichment rule when creating a new collector or for an existing collector. The rule that you need to apply depends on the event source. You can select a rule from the *Categorization rules by event source* table.

   > Event categorization works only when using Kaspersky normalizers.

3. In the KUMA web interface, select the **Resources** → **Active services** section.
4. Select the check box next to the modified collector and click the **Update configuration** button.

The enrichment rule is applied, and the categorization of events is performed.

Categorization rules by event source

System name	Normalizer name	Enrichment rule name
Microsoft, OS event logs	[OOTB] Microsoft Products for KUMA 3	[OOTB] Event categorization. Microsoft Products
Microsoft Sysmon	[OOTB] Microsoft Products for KUMA 3	[OOTB] Event categorization. Microsoft Products
Auditd	[OOTB] Linux auditd syslog for KUMA 3.2	[OOTB] Event categorization. Auditd
Kaspersky Security Center	[OOTB] KSC from SQL	[OOTB] Event categorization. Kaspersky Security Center
Kaspersky Security for Linux Mail Server	[OOTB] KLMS syslog CEF	[OOTB] Event categorization. Kaspersky Security for Linux Mail Server
Kaspersky IoT Secure Gateway	[OOTB] Kaspersky KISG syslog	[OOTB] Event categorization. Kaspersky IoT Secure Gateway
Kaspersky Container Security	[OOTB] Syslog-CEF	[OOTB] Event categorization. Kaspersky Container Security
Kaspersky Industrial CyberSecurity for Networks	[OOTB] Kaspersky Industrial CyberSecurity for Networks 4.2 syslog	[OOTB] Event categorization. Kaspersky Industrial CyberSecurity for Networks
Kaspersky Web Traffic Security	[OOTB] KWTS syslog CEF	[OOTB] Event categorization. Kaspersky Web Traffic Security
Kaspersky Mail Security Gateway	[OOTB] KSMG syslog CEF	[OOTB] Event categorization. Kaspersky Mail Security Gateway

Object attribute

Value of the Object attribute	Description
account	Accounts
configuration	Settings
connection	Network connection
container	Containers
data	Data or tables in the database
dns	DNS queries
email	Email
file	Files
group	Groups
host	Host
http	WEB requests
image	DLLs and drivers
malware	Malicious object
permission	Rights and privileges
process	Process
registry	Registry keys
service	Services and daemons
task	Tasks
VM	Virtual machine
device	Token, USB device, attached device
certificate	Certificate

Action attribute

Value of the Action attribute	Description
access	Request or grant access to an object

add	Create or add
authentication	Authenticate
block	Block, prevent
delete	Delete
detect	Detect or quarantine
end	End of the process
info	Informational event about the object
load	Load object
modify	Modify settings or state of the object
read	Read
receive	Receive
request	Query
send	Send
start	Start of the process
write	Write

Result attribute

Value of the Result attribute	Description
error	Error
failure	Failure
success	Success

Threat attribute

Value of the Threat attribute	Description
malware	Malicious object in the file system
vulnerability	Vulnerability exploits
attack	Attack
ddos	Indicators of a DDoS attack
phishing	Indicators of phishing
c2	Command and Control
discovery	Reconnaissance
policy violation	Violation of security policies
tools	Suspicious tools used
escalation	Privilege escalation

Source type attribute

Value of the Source type attribute	Description
application	Application-level events: logins, configuration changes, errors, and so on.
AV	Events received from anti-virus protection systems.
database	Events from databases related to data manipulation.
IDS	Events from traffic analyzers and network protection tools.
network	Events containing information about network connections coming from firewalls; also events from Netflow, Jflow, and so on.
OS	Events from the operating system, such as Windows, Linux, Cisco IOS, HP-UX, and so on. Standard operating system tools or EDR can be the event provider.

vpn	VPN session events.
web	Events coming from WEB applications and Proxy. Application-level HTTP connection events.
container	Events providing information about the operation of containers.

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 **Access to spaces** section or directly in a user's card. After upgrading to version 4.0, 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. Assign appropriate roles to users.
2. Configure the space sets.

   You can create, edit, or delete space sets. These actions result in audit events.
3. 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 web console, go to the **Settings** → **Access** → **Access to spaces** section.
2. In the opened window, click the **Add** button.
3. In the **Create space sets** 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 web console, go to the **Settings** → **Access** → **Access to spaces** section.
2. In the opened window, click a set of spaces.
3. In the **Edit space sets** 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 web console, go to the **Settings** → **Access** → **Access to spaces** section.
2. In the **Access to spaces** 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**.

You cannot delete the default space set and the 'All spaces' space set.

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 space set.

Grant access to a space set

To grant access to a space set

1. In the KUMA web console, go to the **Settings** → **Access** → **Access to spaces** 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 **Confirm** 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 web console, go to the **Settings** → **Access to spaces** 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 **Confirm** 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 KUMA 4.0 with differentiated access to events

Goal: Update the product.

Action: The administrator updates KUMA to version 4.0.

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 **Access to spaces** 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.

Dashboards

In the **Dashboards** 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.

You can create dashboards manually or use predefined dashboards. If necessary, you can manage the settings of the widgets in the predefined dashboards. The predefined Alerts Overview dashboard is used as the default dashboard.

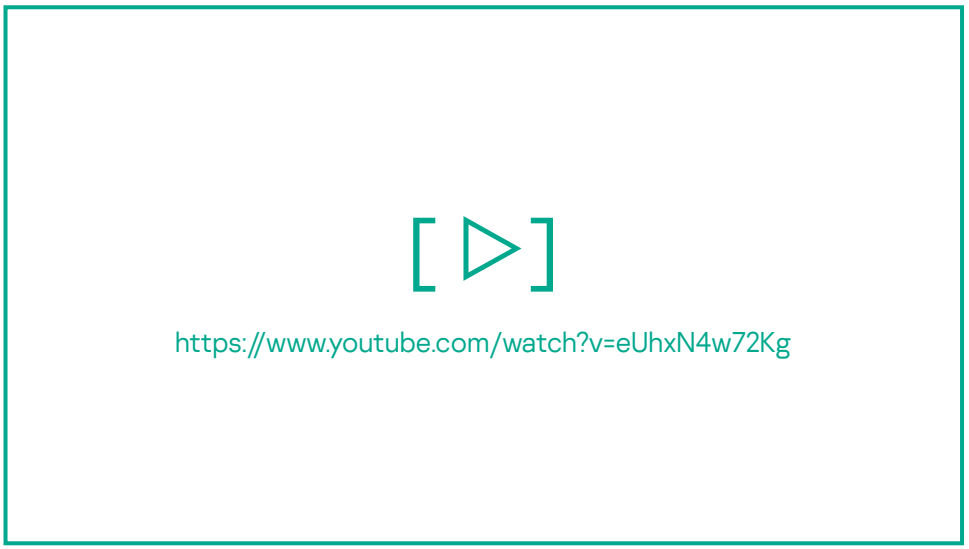
Only users with the Main administrator, Tenant administrator, Tier 2 analyst, and Tier 1 analyst roles can create, manage, or delete dashboards. Users accounts with all roles can view dashboards and set default dashboards. If a dashboard is set as default, that dashboard is displayed for the account every time the user navigates to the **Dashboards** section. The selected default dashboard is saved for the current user account.

The information in the **Dashboards** section is updated in accordance with the schedule configured in dashboards settings. If necessary, you can force the update of the data.

For convenient presentation of information on dashboards, you can enable TV mode. This mode lets you view the dashboards in full-screen mode in FullHD resolution. In TV mode, you can also configure a slide show display for the selected dashboards.

If necessary, you can view a list of all available dashboards for tenants to which they belong in the left part of the window. You can click the **Hide dashboards** button to collapse this list, in which case the widgets of existing dashboards are displayed. You can bring back the list of all dashboards available for tenants by clicking the **Show dashboards** button.

In the list of all dashboards available for tenants, you can also click the add icon + to create folders for grouping dashboards in tenants. You can manage the settings of these folders or delete them by clicking the menu icon ⋮ and selecting an action.



Creating a dashboard

To create a dashboard:

1. Open the KUMA web interface and select the **Dashboards** section.
2. Do one of the following:
   - Click the menu (⋮) icon in the upper-right corner of the window and select **Create dashboard**.
   - Click the **Create dashboard** button under the list of all dashboards.

   This opens the dashboard settings window.
3. In that window, specify a unique name for the dashboard. The name must contain from 1 to 128 Unicode characters and must be unique within the selected tenant.

   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.

4. In the **Dashboard tenant** drop-down list, select the tenant that will own the new dashboard, or a certain folder in the tenant.

   If you had any set of charts from a folder in the tenant open before creating the dashboard, KUMA automatically specifies this tenant when creating the dashboard. You can change the tenant as necessary.

5. In the **Collect data from tenants** drop-down list, select one or more tenants whose data you want to be represented in the dashboard.

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**, select start and end dates of the period in the calendar 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 is populated 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. Once 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 periods available in the **Relative period** list.

     The period is calculated relative to the current time.

   - If you want to specify a **custom period**, set or edit the value of the **Range from** and **Range to** fields 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 period relative to the current time as a formula. You can also combine these methods. 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.

   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 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.

   Kaspersky SIEM stores time values in UTC, but in the web 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 **Refresh interval** drop-down list, select the data refresh period in the dashboard widgets:
   - **No refresh** means data in dashboard widgets are never refreshed.
   - **1 minute**
   - **5 minutes**
   - **15 minutes**
   - **1 hour** (default)
   - **3 hours**
   - **6 hours**
   - **12 hours**
   - **24 hours**
8. If necessary, select tags for the dashboard from the **Tags** drop-down list.
9. If necessary, add a description of the dashboard.
10. Use the **CII status** toggle switch to enable the display of information for assets, alerts, and incidents related to critical information infrastructure (CII). This makes such dashboards viewable only by users whose settings have the **Access to CII facilities** check box selected.

    If this toggle switch is disabled, widgets of the dashboard 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 **Apply**.
12. Click the **Add widget** button and select a widget. This opens a window; in that window, configure the selected widget.

    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 arrow icon ( ) that is displayed when you hover over the lower right corner of the widget.

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`).

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
- Remove widgets
- Download widget data in CSV format

13. Click **Create**.

The new dashboard is created and displayed in the **Dashboards** section of the KUMA web interface.

You can also create a dashboard on the **List** tab in the **Resources** section. To do this, click the **Add** button and select the **Dashboard** resource type.

Selecting a dashboard

To select a dashboard:

1. Expand the list of folders with dashboards in the upper-left corner of the **Dashboards** window.

   Folders with dashboards are organized in accordance with the tenants they belong to.

2. If you want to view the dashboards that you have created, click the **Created by me** button.

   KUMA display only the dashboards that you have created and only those folders in which these dashboards are located. The folders in which your dashboards are located may have been created by other users.

3. Select a folder and a dashboard in that folder.

The selected dashboard is displayed in the **Dashboards** window of the KUMA web interface.

Selecting a dashboard as the default dashboard

To select the dashboard to be displayed by default:

1. In the KUMA web interface, select the **Dashboards** section.
2. Expand the list of folders with dashboards in the upper left corner of the **Dashboards** window.
3. Click the relevant dashboard.

4. Do one of the following:

   - Click the menu icon ⋮ next to the dashboard in the list of folders with dashboards, and in the menu, select **Set as default**.
   - Click the menu icon ⋮ in the upper-right corner of the **Dashboards** window with the dashboard open, and in the menu, select **Set as default**.

The selected dashboard is marked with the **Default** attribute and becomes the default dashboard.

Managing dashboard settings

To manage dashboard settings:

1. In the KUMA web interface, select the **Dashboards** section.
2. Expand the list of folders with dashboards in the upper left corner of the **Dashboards** window.
3. Click the relevant dashboard.
4. Click the menu icon ⋮ next to the dashboard in the list of folders with dashboards, and in the menu, select **Edit**.
5. Make your changes in the window that is displayed. The settings that are available for editing are the same as the settings available when creating a dashboard.
6. Click the **Save** button.

The dashboard settings are modified.

If the dashboard was removed or assigned to a different tenant while you were managing its settings, an error is displayed when you click the **Save** button. The dashboard is not saved. Refresh the KUMA web interface page to see the list of available dashboards in the drop-down list.

Deleting a dashboard

To delete a dashboard:

1. In the KUMA web interface, select the **Dashboards** section.
2. If necessary, click the **Show dashboards** button in the upper-left corner of the **Dashboards** window and open the list of folders with dashboards.
3. Click the relevant dashboard.

4. Do one of the following:

   - Click the menu icon ⋮ next to the dashboard in the list of folders with dashboards, and in the menu, select **Remove**.
   - Click the menu icon ⋮ in the upper right corner of the **Dashboards** window with the dashboard open, and in the menu, select **Remove**.

5. Confirm the removal.

The dashboard is removed.

Enabling and disabling the 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 web interface, select the **Dashboards** section.
2. Click the **TV mode** button in the upper right corner.
3. In the displayed window, move the **TV mode** toggle switch to the **Enabled** position.
4. To configure the dashboard slideshow:
   a. Move the **Slideshow** toggle switch to the **Enabled** position.
   b. In the **Timeout** field, specify how long each dashboard is displayed, in seconds.
   c. Under **Queue**, select the dashboards to display and click **Add to queue**. If no dashboard is selected, the slideshow iterates through all dashboards available to the user.
   d. If necessary, you can delete a dashboard. To do this, select the check box of the dashboard in the queue and click **Delete**.
   e. If necessary, you can change the order of the dashboards by dragging them with the right mouse button.
5. Click the **Save** button.

TV mode will be enabled.

To disnable TV mode:

Click the **Stop** button in the upper right corner of the window where the dashboards are displayed.

TV mode is disabled. The left part of the screen shows a pane containing sections of the KUMA web interface.

If changes are made to the dashboards selected for the slideshow, such changes are automatically reflected in running slideshow sessions.

Setting the period without modifying dashboard settings

When viewing the KUMA dashboard, the period that is specified in the dashboard settings is displayed. If you need to view data for a different period, you can set a period without changing the dashboard settings.

To change the period in the dashboard:

1. In the **Dashboards** section of the KUMA web interface, select a dashboard to display.

   The selected dashboard is displayed in the **Dashboards** window. When opened, the dashboard is generated in accordance with the last scheduled data update.

2. In the **Period** drop-down list above the dashboard, select the absolute or relative period that you are interested in.

3. If you have selected an absolute period, click the **Apply** button.

   The dashboard data is updated.

When changing the period in the dashboard, consider the following:

- The period that you set as described above is not saved in the settings of the dashboard. If you refresh the browser page or close and reopen the dashboard, the display period reverts to the one specified in the dashboard settings.
- Widgets on the dashboard respect the set period. If you specify a different period in the dashboard, the data only changes in widgets for which the **Period** value **as in the dashboard** .
- In TV mode, data is not interactive, therefore the **Period** drop-down list is not displayed and setting a new period is not possible.

Predefined dashboards

KUMA comes with a set of predefined dashboards: The default refresh period for predefined dashboards is **Never**. You can modify these dashboards as necessary.

Dashboard name	Description of widgets in the dashboards
Alerts Overview	• Active alerts—number of alerts that have not been closed. • Unassigned alerts—number of alerts that have the **New** status. • Latest alerts—table with information about the last 10 unclosed alerts belonging to the tenants selected in the dashboard. • Alerts distribution—number of alerts created during the period configured for the widget. • Alerts by severity—number of unclosed alerts grouped by their severity. • Alerts by assignee—number of alerts with the **Assigned** status. The grouping is by account name. • Alerts by status—number of alerts that have the **New**, **Opened**, **Assigned**, or **Escalated** status. The grouping is by status. • Affected users in alerts—number of users associated with alerts that have the **New**, **Assigned**, or **Escalated** status. The grouping is by account name. • Affected assets—table with information about the level of importance of assets and the number of unclosed alerts they are associated with. • Affected assets categories—categories of assets associated with unclosed alerts. • Top event source by alerts number—number of alerts with the **New**, **Assigned**, or **Escalated** status, grouped by alert source (DeviceProduct event field). The widget displays up to 10 event sources. • Alerts by rule—number of alerts with the **New**, **Assigned**, or **Escalated** status, grouped by correlation rules.
Incidents Overview	• Active incidents—number of incidents that have not been closed. • Unassigned incidents—number of incidents that have the **Opened** status. • Latest incidents—table with information about the last 10 unclosed incidents belonging to the tenants selected in the dashboard. • Incidents distribution—number of incidents created during the period configured for the widget. • Incidents by severity—number of unclosed incidents grouped by their severity. • Incidents by assignee—number of incidents with the **Assigned** status. The grouping is by user account name. • Incidents by status—number of incidents grouped by their status. • Affected assets in incidents—number of assets associated with unclosed incidents. • Affected users in incidents—users associated with unclosed incidents. • Affected asset categories in incidents—categories of assets associated with unclosed incidents. • Active incidents by tenant—number of incidents of all statuses, grouped by tenant.
Network 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.
[OOTB] KATA & EDR	• KATA. Top-10 detections by type—visualizes the 10 most common types of events detected by the KATA system. • KATA. Top-10 detections by file type—visualizes the 10 most common file types detected by the KATA system. • KATA. Top-10 user names in detections—visualizes the 10 most common user names detected by the KATA system. • KATA. Top-10 IDS detections—visualizes the 10 most common threats detected by the IDS module of the KATA system. • KATA. Top-10 URL detections—visualizes the 10 most common suspicious URLs detected by the KATA system. • KATA. Top-10 AV detections—visualizes the 10 most common threats detected by the KATA anti-virus module. • EDR. Top-10 MITRE technique detections—visualizes the 10 most common MITRE ATT&CK matrix techniques detected by the EDR system. • EDR. Top-10 MITRE tactic detections—visualizes the 10 most common MITRE ATT&CK matrix tactics detected by the EDR system.
[OOTB] KSC	• KSC. Top-10 users with the most KAV alerts—visualizes the 10 most common user names present in events related to the detection of malicious software, information about which is contained in the KSC system. • KSC. Top-10 most common threats—visualizes the 10 most common types of malware, information about which is contained in the KSC system. • KSC. Number of devices that received AV database updates—visualizes the number of devices on which anti-virus database updates have been installed, information about which is contained in the KSC system. • KSC. Number of devices on which the virus was found—visualizes the number of devices on which malware was detected, information about which is contained in the KSC system. • KSC. Malware detections by hour—visualizes the distribution of the number of malware per hour, information about which is contained in the KSC system.
[OOTB] KSMG	• KSMG. Top-10 senders of blocked emails—visualizes the 10 most common senders of email messages blocked by the KSMG system. • KSMG. Top-10 events by action—visualizes the 10 most common actions performed by the KSMG system. • KSMG. Top-10 events by outcome—visualizes the 10 most common results of actions performed by the KSMG system. • KSMG. Blocked emails by hour—visualizes the distribution of the number of email messages blocked by the KSMG system, by hour.

Dashboard name	Description of widgets in the dashboards
[OOTB] KWTS	• KWTS. Top-10 IP addresses with the most blocked web traffic—visualizes the 10 most common IP addresses from which traffic blocked by the KWTS system originated. • KWTS. Top-10 IP addresses with the most allowed web traffic—visualizes the 10 most common IP addresses from which traffic allowed by the KWTS system originated. • KWTS. Top 10 requests by client application—visualizes the 10 most common applications used to gain access to network resources, as detected by the KWTS system. • KWTS. Top-10 blocked URLs—visualizes the 10 most common URLs from which traffic was allowed by the KWTS system. • KWTS. System action types—visualizes the 10 most common actions performed by the KWTS system. • KWTS. Top-10 users with the most allowed web traffic — visualizes the 10 most common user names of users whose traffic was allowed by the KWTS system.
[OOTB] KSMG files and hashes*	• KSMG. Top-5 blocked hashes — visualizes the 5 most common file hashes in email messages blocked by the KSMG system. • KSMG. Top-5 net-transferred hashes — visualizes the 5 most common "clean" file hashes in email messages tracked by the KSMG system. • KSMG. Top-5 clean file names — visualizes the 5 most common "clean" file names in email messages tracked by the KSMG system. • KSMG. Top-5 blocked files — visualizes the 5 most common file names in email messages blocked by the KSMG system.
[OOTB] KSMG rules and URLs*	• KSMG. Top-5 rules — visualizes the 5 most common triggered rules of the KSMG system. • KSMG. Top-5 URLs — visualizes the 5 most common domains from links in email messages tracked by the KSMG system.
[OOTB] KSMG results*	• KSMG. All results in the last 24 hours — visualizes the hour-by-hour distribution of actions performed by the KSMG system during the last 24-hour period. • KSMG. Top-5 results — visualizes the 5 most common actions performed by the KSMG system.
[OOTB] KSMG e-mail subjects and accounts*	• KSMG. Top-5 e-mail subjects — visualizes the 5 most common subjects of email messages tracked by the KSMG system. • KSMG. Top-5 source accounts — visualizes the 5 most common sender accounts of email messages tracked by the KSMG system. • KSMG. Top-5 destination accounts — visualizes the 5 most common recipient accounts of email messages tracked by the KSMG system.

*Dashboards are available starting from KUMA 3.4.1. Widgets will correctly display information when using the "[OOTB] KSMG 2.1+ syslog CEF" normalizer.

Universal dashboards

A dashboard in KUMA is considered universal if it is created in the Shared tenant. Dashboards created in other tenants are non-universal.

Universal dashboard

The universal dashboard has the following properties:

- The universal dashboard can be viewed by all user roles and in all tenants.
- Data for tenants is displayed in accordance with the general tenant filter.
- Data in widgets on events from the database is obtained in accordance with the space settings configured for the user.

To manage the universal dashboard, the user needs the following roles:

- For creating, configuring, and deleting, the Manage shared resources role is needed. This role rights grant access to all shared resources.
- The Read shared resources role allows viewing the list of universal dashboards. With this role, the user gets read access to all shared resources in the **Resources** section.

  If you have a role that grants access to universal dashboards, you can view the list of universal dashboards in the **Dashboards** and **Resources** sections, but you do not have access to other resources of the Shared tenant.

To view up-to-date data in universal dashboards, you need to subscribe to changes of such dashboards. You can subscribe to all dashboards in the Shared tenant, including the dashboards created by you.

To subscribe to a dashboard in the Shared tenant:

1. In the **Dashboards** section, in the list of dashboards, select and open a dashboard in the Shared tenant.
2. Click the menu ⋮ icon in the upper right corner of the open dashboard.
3. In the menu, select **Subscribe on changes**.

   If necessary, you can unsubscribe by selecting **Unsubscribe from changes** in the list of actions.

If a user's role involves managing CII resources, the display of information on the dashboard takes into account the CII attribute.

Non-universal dashboard

A non-universal dashboard is created in any KUMA tenant except Shared. Such dashboards have the following properties:

- The dashboards can be viewed by users that have access to the tenant in which the dashboard was created.
- Only users that have access to the tenant in which the dashboard was created have access to the data of the widgets in the dashboard.
- If the dashboard has the CII attribute, the data on the dashboard is available to users whose rights allow access to CII.

Exporting dashboards

To export a dashboard:

1. In the **Resources** section, go to the **List** tab.

   This opens a window with a list of all resources available to you in KUMA.

   If necessary, you can configure the display of resources on existing dashboards by selecting **Dashboard** in the **Resource kind** column.
2. Use the check box to select the dashboard to be exported.
3. Click the **Export** button above the table of resources.

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.

   The exported file is saved to the downloads folder of your browser.

Consider the following when exporting dashboards:

- Dashboards are exported without tenants, storage, and space information because these parameters are configured individually for each KUMA deployment.
- Dashboard widgets are also exported without tenant, storage, and space information.
- The parameters of the enrich and lookup SQL functions in widget queries are exported unchanged. If necessary, you can modify these in accordance with your requirements after exporting resources.
- Links to active lists and tables are not exported. If necessary, you can customize these widgets after exporting resources.

Importing dashboards

To import a dashboard:

1. In the **Resources** section, go to the **List** tab.

   This opens a window with a list of all resources available to you in KUMA.

   If necessary, you can configure the display of resources on existing dashboards by selecting **Dashboard** in the **Resource kind** column.

2. Use the check box to select the dashboard 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 dashboard.
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 dashboards:

- If you import a dashboard into the selected tenant and the exportID of the imported dashboard is the same as the exportID of a dashboard that is already present in the tenant, a conflict message is returned with suggestions on how to resolve the conflict. You need to select one of the following options:

  - **Skip**

    If you select **Skip**, all resources selected for import are imported, except the conflicting resource.

  - **Replace**.

    If you select **Replace**, the dashboard is imported and the data is overwritten.

  If you want to look at the difference between dashboards with the same exportID, you can click **Skip**, and then import the dashboard into another tenant and compare the dashboards in different tenants.

- If the name of the imported dashboard is the same as the name of the dashboard that is already present in the tenant, the dashboard is imported and the date and time of the import is appended to the name. For example, Alerts Overview-2025-11-26 12:52:59.

- If the dashboard contains an **Events** widget, we recommend importing the dashboard into the **Shared** tenant. This makes the dashboard universal.

- Importing a dashboard into the Shared tenant requires the **Manage shared resources** role. To view such a dashboard, you need to subscribe to the dashboard.

- If the dashboard is imported into the **Shared** tenant, all storages are automatically specified in the dashboard because access to these is governed by user settings.

- If a dashboard is imported into any tenant other than **Shared**, the tenant of the imported dashboard is specified as the resource tenant, the data source tenant, and the storage tenant. After importing resources, you need to specify spaces for widgets with events and if necessary, expand the list of tenants.

- The parameters of the enrich and lookup SQL functions in widget queries are imported unchanged. If necessary, you can modify these in accordance with your requirements after importing resources.

- Links to active lists and tables are not imported. If necessary, you can customize these widgets after importing resources.

- If the dashboard contains the **Active lists** or the **Context tables** widget, we recommend duplicating such dashboards and editing the settings of the duplicates to adapt them to your infrastructure: specifying a correlator, selecting active lists and context tables.

Reports

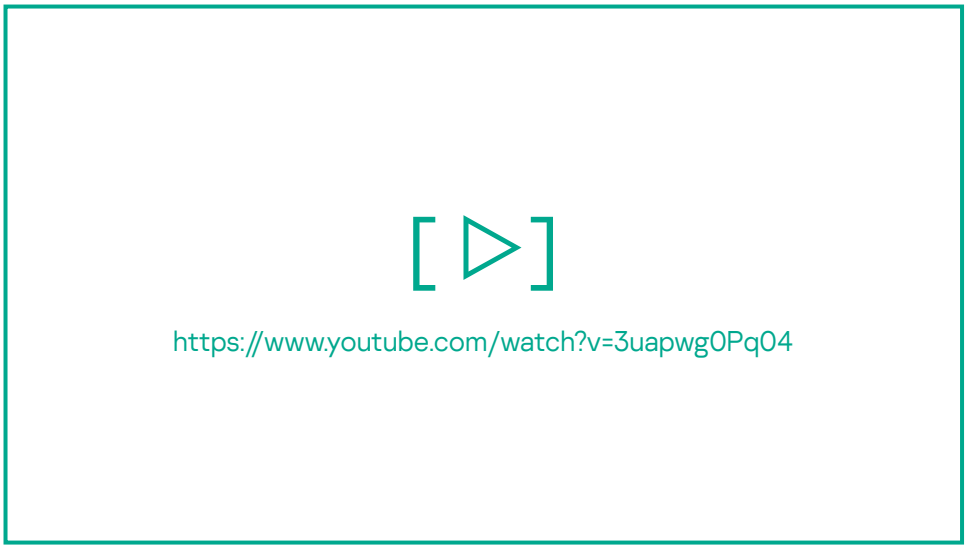
You can schedule the generation of process and object status reports in KUMA.

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 ⌄ 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.

  > Regular expressions are used when searching for report templates.

- **Tenant** is 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 ⌄ in the column header and selecting **Ascending** or **Descending**.

- **Created by** is 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** is 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** is 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. In the web interface, perform one of the following actions:

   - Under **Reports** → **Templates**, in the toolbar or menu ⋮, click **Create**.
   - Under **Resources** → **List**, in the toolbar or menu ⋮, click **New** and select **Report template**.

   The pane for creation a report template opens on the right.

2. In the **Template name** field, enter a name for the report template. The name must contain from 1 to 128 Unicode characters and must be unique within the selected tenant.

   > The names of custom report templates must not match the names of predefined report templates in the Main tenant.

   > When trying 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 report template, the Main tenant returns an error. In other tenants you can create or duplicate a report template with the same name as a predefined report template. When creating report templates, we recommend using unique names.

3. In the **Template tenant** drop-down list, select the tenant that will own the layout being created.

4. In the **Collect data from tenants** drop-down list, select one or more tenants whose data you want to use for the reports.

5. 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**, select start and end dates of the period in the calendar 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 is populated 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. Once 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 periods available in the **Relative period** list.

     The period is calculated relative to the current time.

   - If you want to specify a **custom period**, set or edit the value of the **Range from** and **Range to** fields 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 period relative to the current time as a formula. You can also combine these methods. 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.

By default, the **1 day** (`now-1d`) relative period is selected. The bounds of the period are included. For example, for **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 **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.

Kaspersky SIEM stores time values in UTC, but in the web 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.

6. 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.

7. If necessary, select tags for the report template from the **Tags** drop-down list.

8. If necessary, add a description of the report template.

9. 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.

10. Click **Save**.

    The report modification window opens.

11. If necessary, add the logo of the report template by dragging the image into the field or by clicking the **Upload a file** link. The image must be in JPG, PNG or GIF format. The maximum size of an image is 3 MB.

    The added logo will be displayed in the report instead of the solution logo.

12. Click the **Add widget** button and select a widget. This opens a window; in that window, configure the selected widget.

    You can add multiple widgets. You can drag widgets using the handle icon ( ) in the widget's upper left corner. To resize the widget, use the stretch icon ( ) that appears when you point to the lower right corner of the widget.

    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`).

    By clicking the menu icon in the upper right corner of the widget, you can change the settings of the widgets, copy and delete widgets.

13. Click **Create**.

    A new report template is created. You can generate this report manually or schedule automatic report generation.

Configuring report schedule

You can configure a schedule for sending reports to the specified email addresses.

If the report being sent contains only widgets with errors, such a report is not sent. If the report contains at least one widget without errors, such a report is sent.

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: HTML, PDF, 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.

3. Make the necessary changes:

- Change the name of the report template. It must be unique within the selected tenant.

  Custom report templates cannot have the same names as predefined report templates.

  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.

- 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 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 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 the **Resources** → **List** 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. In the toolbar or menu (⋮), click **Export**.

3. 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.

4. 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.
- If before exporting you have deleted the predefined report template and created a custom report template with the same name as predefined report template, the date and time of export is appended to the name of the report template. For example, Incidents Overview-2025-11-26 12:52:59.

To import a report template:

1. Open the KUMA web interface and go to the **Resources** → **List** 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. In the toolbar or menu (⋮), click **Import**.

3. In the sidebar, in the **Tenant** drop-down list, select the tenant to which you want to import the report template.

4. Select the import source: **File** or **Repository**.

5. 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.

6. Click the **Import** button.

7. If necessary, resolve conflicts between the resources imported from the file and the existing resources.

Consider the following when importing report templates:

- If you import a report template, and the exportID of the imported report template is the same as the exportID of a report template that is already present in the tenant, a conflict message is returned with suggestions on how to resolve the conflict. You need to select one of the following options:

  - **Skip**

    If you select **Skip**, all resources selected for import are imported, except the conflicting resource.

  - **Replace**.

    If you select **Replace**, the report template is imported and the data is overwritten.

  If you want to look at the difference between report templates with the same exportID, you can click **Skip**, and then import the report template into another tenant and compare the report templates in different tenants.

- If the name of the imported report template is the same as the name of the report template that is already present in the tenant, the report template is imported and the date and time of the import is appended to the name. For example, Incidents Overview-2025-11-26 12:52:59.

- 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.
- Widge 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 on the **Generated reports** tab in 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.

   This opens a window with widgets that display the analytics of the reports. If a widget displays information about events, alerts, incidents, active lists, or context tables, 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.

If the report in CSV or Excel format contains widgets that share the same name, the widget files are automatically renamed by adding a numeric suffix. Renaming is automatic, therefore, when the same report is saved to different formats, suffixes of renamed files of the same widget may not be the same across formats. For example, a `widget(2)` file in one format does not necessarily correspond to the `widget(2)` file in another format. For convenience, we recommend giving widgets unique names, regardless of the report format.

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 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.

Widgets

Widgets let you monitor the operation of the solution. Widgets are organized into widget groups, each one related to the analytics type they provide. The following widget groups and widgets are available in KUMA:

- **Events**—widget for creating analytics based on events.
- **Active lists**—widget for creating analytics based on active lists of correlators.
- **Alerts**—group for alert analytics.

  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 the **New** status.
  - **Alerts by assignee**—number of alerts with the **Assigned** status, grouped by account name.
  - **Alerts by status**—number of alerts that have the **New**, **Opened**, **Assigned**, or **Escalated** status, grouped by status.
  - **Alerts by severity**—number of unclosed alerts grouped by their severity.
  - **Alerts by rule**—number of unclosed alerts grouped by correlation rule.
  - **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.

- Assets—group for analytics for assets from processed events. This group includes the following widgets:
  - **Affected assets**—table with information about the level of importance of assets and the number of unclosed alerts they are associated with.
  - **Affected asset categories**—categories of assets linked to unclosed alerts.
  - **Number of assets**—number of assets that were added to KUMA.
  - **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 incident analytics.

  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 assignee**—number of incidents with the **Assigned** status, grouped by user account name.
  - **Incidents by status**—number of incidents grouped by status.
  - **Incidents by severity**—number of unclosed incidents grouped by their severity.
  - **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 in incidents**—number of assets associated with unclosed incidents.
  - **Affected assets categories in incidents**—asset categories associated with unclosed incidents.
  - **Affected users in Incidents**—users associated with incidents.
  - **Latest incidents**—table with information about the last 10 unclosed incidents belonging to the tenants selected in the layout.

- **Event sources**—group for event source analytics. The group includes the following widgets:
  - **Top event sources (value in DeviceProduct field) by number of base events collected in correlation events** displays the number of base events collected in correlation events and grouped by data sources specified in the DeviceProduct field.
  - **Top event sources (value in DeviceProduct field) by convention rate** displays the ratio of the number of base events in correlation events and the total number of base events grouped by data sources specified in the DeviceProduct field.

    In some cases, the number of correlation events 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.
  - **Number of AD users**—number of Active Directory accounts received via LDAP during the period configured for the widget.
- Image—a widget for displaying an uploaded image on the dashboard or in the report.
- Text—a widget for displaying arbitrary text on the dashboard or in the report.

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.

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
- Speedometer 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 the KUMA web interface.

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 30д). 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 the KUMA web interface 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 the request returns data subdivided into a large number of sectors, then the **Other** sector is added to optimize the display of the widget. You can make the **Other** sector smaller by making a more detailed query. You can use `LIMIT` and `WHERE` clauses to refine your query.

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 the 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 the KUMA web interface 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 the request returns data subdivided into a large number of bars, then the **Other** bar is added to optimize the display of the widget. You can make the **Other** bar smaller by making a more detailed query. You can use `LIMIT` and `WHERE` clauses to refine your query.

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 the KUMA web interface 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 of a query to display a line chart with multiple lines

```
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 the KUMA web interface 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 the KUMA web interface 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 on a dashboard while creating or editing the dashboard.

To create a widget:

1. Create a dashboard or go to edit mode of the selected dashboard.
2. Click **Add widget**.
3. Select a widget type from the drop-down list.
   This opens the widget settings window.
4. Edit the widget settings.
5. If you want to see how the data will be displayed in the widget, click **Preview**.

6. Click **Save** in the widget settings window.
   The widget appears on the dashboard.
7. Click the **Save** button in the dashboard settings window.

Editing a widget

To edit a widget:

1. In the KUMA web interface, select the **Dashboards** section.
2. Expand the list of dashboards in the upper left corner of the window.
3. Select a dashboard and click the **Edit** button.
   This opens the dashboard settings window.
4. On the widget that you want to edit, click the menu (⋮) icon.
5. Select **Edit**.
   This opens the widget settings window.
6. Edit the widget settings.
7. Click **Save** in the widget settings window.
8. Click the **Save** button in the dashboard settings window.

The changes are applied to the widget.

Deleting a widget

To delete a widget:

1. In the KUMA web interface, select the **Dashboards** section.
2. Expand the list of dashboards in the upper left corner of the window.
3. Select a dashboard and click the **Edit** button.
   This opens the dashboard settings window.
4. On the widget that you want to delete, click the menu icon ⋮.
5. Select **Delete**.
6. This opens a confirmation window; in that window, click **OK**.
7. 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
Name	Name of the widget. When saving a report in Excel format, the name of the widget displayed in the header of the sheet is limited to 31 characters. We recommended keeping this limitation in mind when naming widgets.
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.
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.

Setting	Description
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 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.

Setting	Description
	• 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**, select start and end dates of the period in the calendar 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 is populated 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. Once 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 periods available in the **Relative period** list.

    The period is calculated relative to the current time.

  - If you want to specify a **custom period**, set or edit the value of the **Range from** and **Range to** fields 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 period relative to the current time as a formula. You can also combine these methods. 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.

Kaspersky SIEM stores time values in UTC, but in the web 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.

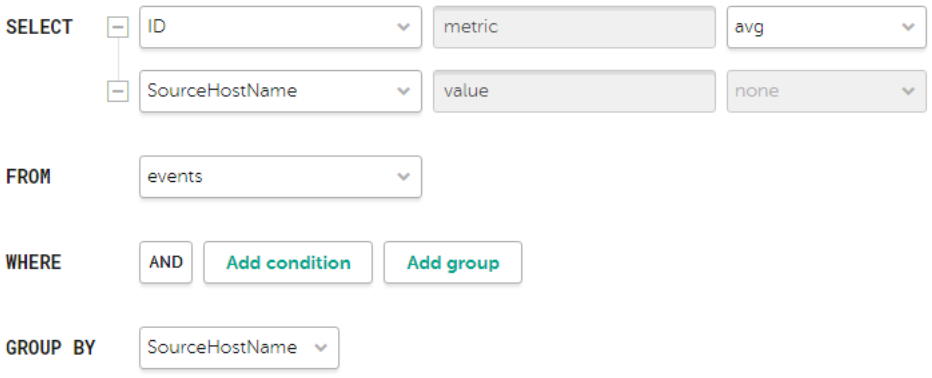
e. **ORDER BY**—columns used as the basis for sorting the returned data. This parameter is not available for the **Date Histogram** and **Counter** graph types.

- 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
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	Displays the sum total of values. The rounding of the value is determined by the **Decimals** setting. This setting is available for **Pie chart**, **Bar chart**, and **Date Histogram** chart types. The toggle switch is turned off by default.
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
Name	Name of the widget. When saving a report in Excel format, the name of the widget displayed in the header of the sheet is limited to 31 characters. We recommended keeping this limitation in mind when naming widgets.
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.
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. The configuration process involves the following steps:

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. This 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
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	Displays the sum total of values. The rounding of the value is determined by the **Decimals** setting. This setting is available for **Pie charts** and **Bar charts**. The toggle switch is turned off by default.
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 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.

Setting	Description
Name	Name of the widget. When saving a report in Excel format, the name of the widget displayed in the header of the sheet is limited to 31 characters. We recommended keeping this limitation in mind when naming widgets.
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	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.
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 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. • 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 chart 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 backticks: ", `. When using spaces or non-Latin 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. 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`.

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. The configuration process involves the following steps:

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. This 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

Setting	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

Settings	Description
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	Displays the sum total of values. The rounding of the value is determined by the **Decimals** setting. This setting is available for **Pie charts** and **Bar charts**. The toggle switch is turned off by default.
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 custom widget

You can use the **Assets → Customized widget** to get advanced asset analytics from processed events using the query builder. In the query, you must 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. When saving a report in Excel format, the name of the widget displayed in the header of the sheet is limited to 31 characters. We recommended keeping this limitation in mind when naming widgets.
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.

Image widget

This widget allows adding an image to a dashboard or a report. For example, you can use the **Image** widget to brand a dashboard or report with your company logo.

The **Image** widget does not have a sidebar with settings.

To add an image file, hover over the widget and click **Upoad image**; in the file selection window, select a file, and after adding the file to the widget, click **Save**.

You can add multiple **Image** widgets to your dashboard. You can move the widgets around your dashboard and rearrange them by dragging the (⠿) button.

You can create a dashboard with an **Image** widget without adding an image file, for example, to design a layout. KUMA will save the dashboard, and you can edit it later by clicking the **Edit** button on the toolbar, and add an image.

Minimum widget dimensions: 225x161 pixels. Maximum dimensions: 465x692 pixels. You can click the (⤡) button to resize the image in the widget. When resizing, the image is not cropped, but scaled relative to the size of the widget.

Maximum image file size: 3 MB.

Supported image file formats: xbm, tif, pjp, apng, svgz, jpg, jpeg, heif, ico, tiff, webp, jfif, heic, gif, svg, png, bmp, pjpeg, avif.

Downloading widget details in CSV format is not supported.

You can create, edit, duplicate, or delete this widget.

Text widget

This widget allows adding arbitrary text to a dashboard or a report. For example, you can use the **Text** widget to display a title or a brief description of the dashboard or report.

To add or edit text, click on the widget. This activates a text box, and a panel appears that allows formatting the text. You can enter up to 2000 characters.

The **Text** widget does not have a sidebar with settings. A widget of this type cannot be downloaded in CSV format.

Other widgets

This section describes the settings for all widgets, except for 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 (4).
- Table (▦).
- Bar chart (☰).
- Date Histogram (📊).
- Line chart
- Stacked Bar chart
- Speedometer 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. When saving a report in Excel format, the name of the widget displayed in the header of the sheet is limited to 31 characters. We recommended keeping this limitation in mind when naming widgets.
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	Displays the sum total of values. The rounding of the value is determined by the **Decimals** setting. The toggle switch is turned off by default.
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.
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.

Setting	Description
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, Stacked 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-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**.
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.
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.

Setting	Description
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 enlarge the widget to better fit the data.
Show total	Displays the sum total of values. The rounding of the value is determined by the **Decimals** setting. The toggle switch is turned off by default.
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.
Period segments length	Duration of the segments into which you want to divide the period. Available for graphs of the **Date Histogram** type.

Settings of Speedometer charts

The table below lists the settings of 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.

Setting	Description
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**, select start and end dates of the period in the calendar 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 is populated 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. Once 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 periods available in the **Relative period** list.

    The period is calculated relative to the current time.

  - If you want to specify a **custom period**, set or edit the value of the **Range from** and **Range to** fields 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 period relative to the current time as a formula. You can also combine these methods. 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.

Kaspersky SIEM stores time values in UTC, but in the web 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 **Event** 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 on 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`. 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)`.

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 widgets of the "Active list" type

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.

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.

Working with alerts

Alerts are created when a sequence of events is received that triggers a correlation rule. You can find more information about alerts in this section.

In the **Alerts** section of the KUMA web interface, you can view and process the alerts registered by the solution. Alerts can be filtered. When you click the alert name, a window with its details opens.

The alert date format depends on the localization language selected in the solution settings. Possible date format options:

- English localization: YYYY-MM-DD.
- Russian localization: DD.MM.YYYY.

Alert life cycle

Below is the life cycle of an alert:

1. KUMA creates an alert when a correlation rule is triggered. The alert is named after the correlation rule that generated it. Alert is assigned the **New** status.

   > Alerts with the **New** status continue to be updated with data when correlation rules are triggered. If the alert status changes, the alert is no longer updated with new events, and if the correlation rule is triggered again, a new alert is created.

2. A security officer assigns the alert to an operator for investigation. The alert status changes to **assigned**.
3. The operator performs one of the following actions:
   - Close the alert as false a positive (alert status changes to **closed**).
   - Respond to the threat and close the alert (alert status changes to **closed**).
   - Creates an incident based on the alert (the alert status changes to **In incident**).

Filling alerts with events

By default, the number of events associated with an alert cannot exceed 10,000. When the number of events associated with the alert reaches 10,000, a new alert is created with the same name, but with a new ID. You can view the alert ID in the alert details. If necessary, you can change this value in the properties of the segmentation rule in the **Correlation events limit** field. For this to work, the segmentation rule must be enabled and linked to the alert correlation rules of the tenant.

Alert segmentation

Using the segmentation rules, the stream of correlation events of the same type can be divided to create more than one alert.

Configuring alerts table

The main part of the **Alerts** section shows a table containing information about registered alerts.

The following columns are displayed in the alerts table:

- Severity ( )—shows the importance of a possible security threat: Critical , High , Medium , or Low .
- **Name**—alert name.
- **Status**—current status of an alert:
  - **New**—a new alert that hasn't been processed yet.
  - **Assigned**—the alert has been processed and assigned to a security officer for investigation or response.
  - **Closed**—the alert was closed. Either it was a false alert, or the security threat was eliminated.
  - **Escalated**—an incident was generated based on this alert.
- **Assigned to**—the name of the security officer the alert was assigned to for investigation or response.
- **Incident**—name of the incident to which this alert is linked.
- **Additional information** is the value of the field selected for display in the **Alert information** section when configuring alert filling. When you select some fields (for example, **SourceAddress**, **DestinationAddress**, **SourceUserName**, **DestinationUserName**), their associated fields are additionally displayed (**SourceAssetName**, **DestinationAssetName**, **SourceAccountName**, **DestinationAccountName**). The values of these fields are combined displayed in the **Additional information** column, separated by commas, in the following order: first the selected field, then the additional field.

  > For the selected field, the earliest recorded and non-empty value of the event field associated with the alert is displayed.

  By default, this column is not displayed in the table. To add it, click the gear icon ( ) and select the check box next to the column title. .
- **First seen**—the date and time when the first correlation event of the event sequence was created, triggering creation of the alert.
- **Last seen**—the date and time when the last correlation event of the event sequence was created, triggering creation of the alert.
- **Categories**—categories of alert-related assets with the highest severity. No more than three categories are displayed.

- **Tenant**—the name of the tenant that owns the alert.
- **CII**—an indication whether the related to the alert assets are the CII objects. The column is hidden from the users who do not have access to CII objects.

You can view the alert filtering tools by clicking the column headers. When filtering alerts based on a specific parameter, the corresponding header of the alerts table is highlighted in yellow.

You can click the gear icon (⚙), to customize the columns displayed in the alerts table.

In the **Search** field, you can use regular expressions to find alerts by their linked assets, users, tenants, or correlation rules. Parameters that can be used for a search:

- Assets: name, FQDN, IP address.
- Active Directory accounts: attributes displayName, SAMAccountName, and UserPrincipalName.
- Correlation rules: name.
- KUMA users who were assigned alerts: name, login, email address.
- Tenants: name.

Filtering alerts

In KUMA, you can perform alert selection by using the filtering and sorting tools in the **Alerts** section.

The filter settings can be saved. Existing filters can be deleted.

Saving and selecting an alert filter

In KUMA, you can save changes to the alert table settings as filters. Filters are saved on the KUMA Core server and are available to all KUMA users of the tenant for which they were created.

To save the current filter settings:

1. In the **Alerts** section of KUMA open the **Filters** drop-down list.
2. Select **Save current filter**.

   A field will appear for entering the name of the new filter and selecting the tenant that will own it.
3. Enter a name for the filter. The name must be unique for alert filters, incident filters, and event filters.
4. In the **Tenant** drop-down list, select the tenant that will own the filter and click **Save**.

The filter is saved.

To select a previously saved filter:

1. In the **Alerts** section of KUMA open the **Filters** drop-down list.
2. Select the relevant filter.

   To select the default filter, put an asterisk to the left of the relevant filter name in the **Filters** drop-down list.

The filter is selected.

To reset the current filter settings,

Open the **Filters** drop-down list and select **Clear filters**.

Deleting an alert filter

To delete a previously saved filter:

1. In the **Alerts** section of KUMA open the **Filters** drop-down list.
2. Click next to the configuration that you want to delete.
3. Click **OK**.

The filter is deleted for all KUMA users.

Viewing details on an alert

To view details on an alert:

1. In the solution's web interface window, select the **Alerts** section.

   The alerts table is displayed.
2. Click the name of the alert whose details you want to view.

This opens a window containing information about the alert.

The upper part of the alert details window contains a toolbar and shows the alert severity and the user name to which the alert is assigned. In this window, you can process the alert: change its severity, assign it to a user, and close and create an incident based on the alert.

Details on alert section

This section lets you view basic information about an alert. It contains the following data:

- **Correlation rule severity** is the severity of the correlation rule that triggered the creation of the alert.
- **Max asset category severity**—the highest severity of an asset category assigned to assets related to this alert. If multiple assets are related to the alert, the largest value is displayed.

- **Linked to incident**—if the alert is linked to an incident, the name and status of the alert are displayed. If the alert is not linked to an incident, the field is blank.
- **First seen**—the date and time when the first correlation event of the event sequence was created, triggering creation of the alert.
- **Last seen**—the date and time when the last correlation event of the event sequence was created, triggering creation of the alert.
- **Alert ID**—the unique identifier of an alert in KUMA.
- **Tenant**—the name of the tenant that owns the alert.
- **Correlation rule**—the name of the correlation rule that triggered the creation of the alert. The rule name is represented as a link that can be used to open the settings of this correlation rule.

This section can additionally display the field of the correlation event and its value selected from the Display in Alert information drop-down list when **configuring alert filling**. For some of the fields, when they are selected, their associated fields are additionally displayed. For example, for the **SourceAddress** field, the additionally displayed field is **SourceAssetName**, for **DestinationAddress** it is **DestinationAssetName**, for **DeviceAddress** it is **DeviceAssetName**, for **SourceHostName it is SourceAssetName**, for **DestinationHostName** it is **DestinationAssetName**, for **DeviceHostName** it is **DeviceAssetName**. When you select the **SourceUserName** field, the **SourceAccountName** field is additionally displayed, and when you select **DestinationUserName**, **DestinationAccountName** is displayed. This works only in one direction, that is, for example, selecting the **DestinationAccountName** field does not display the **DestinationUserName** field.

> For the selected field, the earliest recorded and non-empty value of the event field associated with the alert is displayed.

Related events section

This section contains a table of events related to the alert. If you click the arrow (›) icon near a correlation rule, the base events from this correlation rule will be displayed. Events can be sorted by severity and time.

Selecting an event in the table opens the details area containing information about the selected event. The details area also displays the **Detailed view** button, which opens a window containing information about the correlation event.

The **Find in events** links below correlation events and the **Find in events** button to the right of the section heading are used to go to alert investigation.

You can use the **Download events** button to download information about related events into a CSV file (in UTF-8 encoding). The file contains columns that are populated in at least one related event.

Some CSV file editors interpret the separator value (for example, \n) in the CSV file exported from KUMA as a line break, not as a separator. This may disrupt the line division of the file. If you encounter a similar issue, you may need to additionally edit the CSV file received from KUMA.

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.

Related endpoints section

This section contains a table of assets related to the alert. Asset information comes from events that are related to the alert. You can search for assets by using the **Search for IP addresses or FQDN** field. Assets can be sorted using the **Count** and **Endpoint** columns.

This section also displays the assets related to the alert. Clicking the name of the asset opens the **Asset details** window. In that window, in the **Related alerts** section, in the **Find related alerts** drop-down list, you can select the period for which you want to find related alerts: last day, week, two weeks (default value), month, or year. After you select the period, the number of related alerts found is displayed. To view the full list, click the **Find in Alerts** button, to open the **Alerts** page with a filtered list of found related alerts.

You can use the **Download assets** button to download information about related assets into a CSV file (in UTF-8 encoding). The following columns are available in the file: **Count**, **Name**, **IP address**, **FQDN**, **Categories**.

Related users section

This section contains a table of users related to the alert. User information comes from events that are related to the alert. You can search for users using the **Search for users** field. Users can be sorted by the **Count**, **User**, **User principal name** and **Email** columns.

You can use the **Download users** button to download information about related users into a CSV file (in UTF-8 encoding). The following columns are available in the file: **Count**, **User**, **User principal name**, **Email**, **Domain**, **Tenant**.

Change log section

This section contains entries about changes made to the alert by users. Changes are automatically logged, but it is also possible to add comments manually. Comments can be sorted by using the **Time** column.

If necessary, you can enter a comment for the alert in the **Comment** field and click **Add** to save it.

Changing alert name

To change the alert name:

1. In the KUMA web interface window, select the **Alerts** section.

   The alerts table is displayed.

2. Click the name of the alert whose details you want to view.

   This opens a window containing information about the alert.

3. In the upper part of the window, click ✎ and in the field that opens, enter the new name of the alert. To confirm the name, press **ENTER** or click outside the entry field.

Alert name is changed.

Configuring alert filling

To configure alert filling:

1. In the KUMA web interface, go to the **Settings** → **Other** → **Alerts** section.

2. In the **Alerts** window, go to the **Alerts filling** tab and manage the following settings:

   a. In the **Status** drop-down list, select the status that an alert must have to be filled with events. The default setting is **New**. You can select multiple values. The **New** value cannot be removed from the selection. Available values:

   - **New** means the created correlation events continue to be linked to the alert only if the alert status is **New**.
   - **In incident** means the created correlation events continue to be linked to the alert only if the alert status is **In incident**. If the status of the alert changes to **In incident**, it can revert to the **New** status if the alert is unlinked from the incident. In that case, the filling of the alert with events stops.

     Example: the alert has the **In incident** status. While in the **In incident** status, no new events were received. If you unlink the alert from the incident, the alert will be filled with events. If, while the alert was linked to the incident, more events were received and a new alert was created, the last created alert is filled with events.

   - **Assigned** means the created correlation events continue to be linked to the alert only if the alert status is **Assigned**.

   b. In the **Alert was created no later than (hours)** field, specify the time during which the alert will be filled with events. The value must be an integer. The time is counted from the moment the alert is created. Default value: 336 hours (2 weeks). The minimum value is 1 hour.

If some values are specified for the **Status**, **Alert was created time (hours)** and **Maximum number of events** settings, the filling of the alert is stopped when the first of the following conditions is satisfied:

- The alert status has changed and no longer matches the selected values.
- The specified filling time has run out.
- The number of events in the alert has reached the value specified in the **Maximum number of events** field.

If, after the alert has stopped filling, it regains a status in which filling is allowed, no new events are added to it. For example, if filling is configured only for the **New** status, filling stops after the alert transitions to the **In incident** status. If the alert then regains the **New** status, no new events are added to it.

c. In the **Event storage** drop-down list, select the storage where the alert events are stored. Only one storage can be specified. The list shows storage resources, instead of running services. For alerts to be filled with events, the storage service must be running.

Alerts created before specifying the storage in the filling settings do not contain any events. After the **Event storage** setting is specified or changed, existing alerts are not filled with events from the specified storage. In that case, new alerts are created with events from the specified storage. After selecting an event storage, this setting can take up to two minutes to be applied to alerts. Alerts created before this parameter is applied may not contain events from the specified storage.

d. In the **Display in Alert details** drop-down list, select the field of the correlation event to be displayed in the alert card and in the **Additional info** column in the **Alerts** section. You can select one of the following values:

- **DestinationAddress** is the IPv4 or IPv6 address of the destination (asset) involved in the event.
- **SourceAddress** is the IPv4 or IPv6 address of the device that was the source of the event. If the user logged in using a proxy server, the address of the proxy server is specified.
- **DeviceAddress** is the IPv4 or IPv6 address of the device from which the event was received.
- **DestinationAssetName** is the name of the asset at the destination.
- **SourceAssetName** is the name of the asset at the event source.
- **DeviceAssetName** is the name of the asset on the device from which the event was received.
- **DestinationHostName** is the domain name of the destination.
- **SourceHostName** is the domain name of the event source.
- **DeviceHostName** is the name of the device from which the event was received. FQDN of the device, if available.
- **DestinationUserName** is the user name of the destination.
- **SourceUserName** is the user name of the source.
- **DestinationAccountName** is the name of the account of the destination.
- **SourceAccountName** is the name of the account of the event source.

- **Message** is the brief description of the event.
- **DeviceCustomString1**, **DeviceCustomString2**, **DeviceCustomString3**, **DeviceCustomString4**, **DeviceCustomString5**, **DeviceCustomString6** are fields for mapping the string value.

You can select only one field. For some fields, an additional associated field is automatically displayed. If you select a field that refers to an address or host name, the associated **AssetName** field is additionally displayed. If you selected a field associated with a user name, the **AccountName** field is additionally displayed. In these cases, the field values are combined with a comma: the selected field comes first and is followed by the additional field.

The modified setting applies only to new alerts.

e. In the **Maximum number of events** field, specify the maximum number of events that can be added to one alert. After the specified amount is reached, filling of the alert stops.

The value must be an integer. The default value is `10000`. The minimum value is `1`. No maximum value is set.

When you save segmentation rules, the value of the number of events is checked. If the number of events specified in the rule is greater than the value of the Maximum number of events setting, the saving process finishes with an error. For segmentation rules created before the value of the setting was decreased, the change is not applied. Such rules continue to work, but the number of events added to an alert never exceeds the value of the **Maximum number of events** setting.

Alert filling is configured.

Processing alerts

You can change the alert severity, assign an alert to a user, close the alert, or create an incident based on the alert.

To process an alert:

1. Select required alerts using one of the methods below:

   - In the **Alerts** section of the KUMA web interface, click the alert whose information you want to view.

     The Alert window opens and provides an alert processing toolbar at the top.

   - In the **Alerts** section of the KUMA web interface, select the check box next to the required alert. It is possible to select more than one alert.

     > Alerts with the **closed** status cannot be selected for processing.

     A toolbar will appear at the bottom of the window.

2. If you want to change the severity of an alert, select the required value in the **Severity** drop-down list:

   - **Low**
   - **Medium**
   - **High**
   - **Critical**

   The severity of the alert changes to the selected value.

3. If you want to assign an alert to a user, select the relevant user from the **Assign to** drop-down list.

   You can assign the alert to yourself by selecting **Me**.

   The status of the alert will change to **Assigned** and the name of the selected user will be displayed in the **Assign to** drop-down list.

4. In the **Related users** section, select a user and configure Active Directory response settings.

   a. After the related user is selected, in the **Account details** window that opens, click **Response via Active Directory**.

   b. In the **AD command** drop-down list, select one of the following 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

   - Block account

   c. Click **Apply**.

5. If required, create an incident based on the alert:

   a. Click **Create incident**.

      The window for creating an incident will open. The alert name is used as the incident name.

   b. Update the desired incident parameters and click the **Save** button.

   The incident is created, and the alert status is changed to **Escalated**. An alert can be unlinked from an incident by selecting it and clicking **Unlink**.

6. If you want to close the alert:

   a. Click **Close alert**.

      A confirmation window opens.

b. Select the reason for closing the alert:

- **Responded**. This means the appropriate measures were taken to eliminate the security threat.
- **Incorrect data**. This means the alert was a false positive and the received events do not indicate a security threat.
- **Incorrect correlation rule**. This means the alert was a false positive and the received events do not indicate a security threat. The correlation rule may need to be updated.

c. Click **OK**.

The status of the alert is changed to **Closed**. Alerts with this status are no longer updated with new correlation events and aren't displayed in the alerts table unless the **Closed** check box is selected in the **Status** drop-down list in the alerts table. You cannot change the status of a closed alert or assign it to another user.

Alert investigation

Alert investigation is used when you need to find more information about the threat that triggered the alert: is the threat real, what is its origin, what elements of the network environment are affected by it, how should the threat be dealt with. Studying the events related to the correlation events that triggered an alert can help you determine the course of action.

For convenience of investigating alerts, make sure that time is synchronized on all devices involved in the event life cycle (event sources, KUMA servers, client hosts) with the help of Network Time Protocol (NTP) servers.

The alert investigation mode is enabled in KUMA when you click the **Find in events** link in the alert window or the correlation event window. When the alert investigation mode is enabled, the events table is shown with filters automatically set to match the events from the alert or correlation event. The filters also match the time period of the alert duration or the time when the correlation event was registered. You can change these filters to find other events and learn more about the processes related to the threat.

An additional drop-down list becomes available in alert investigation mode:

- **All events**—view all events.
- **Related to alert** (selected by default)—view only events related to the alert.

When filtering events related to an alert, there are limitations on the complexity of SQL search queries.

You can manually assign an event of any type except the correlation event to an alert. Only events that are not related to the alert can be linked to it.

You can create and save event filters in alert investigation mode. When using this filter in normal mode, all events that match the filter criteria are selected regardless of whether or not they are related to the alert that was selected for alert investigation.

To link an event to an alert:

1. In the **Alerts** section of the KUMA web interface, click the alert that you want to link to the event.

   The Alert window opens.

2. In the **Related events** section, click the **Find in events** button.

   The events table is opened and displayed with active date and time filters matching the date and time of events linked to the alert. The columns show the settings used by the correlation rule to generate the alert. The **Link to alert** column is also added to the events table showing the events linked to the alert.

3. In the drop-down list select **All events**.

4. If necessary, modify the filters to find the event that you need to link to the alert.

5. Select the relevant event and click the **Link to alert** button in the lower part of the event details area.

   The event will be linked to the alert. You can unlink this event from the alert by clicking in the **Unlink from alert** detailed view.

When an event is linked to or unlinked from an alert, a corresponding entry is added to the **Change log** section in the Alert window. You can click the link in this entry to open the details area and unlink or link the event to the alert by clicking the corresponding button.

Retention of alerts and incidents

The alert and incident retention conditions are determined by the following criteria:

- Retention period. By default, KUMA retains alerts and incidents for 365 days, but you can set a different period by changing the **Alert retention period, days** setting.
- Deletion conditions. After the expiration of the specified retention period, if certain conditions are satisfied, alerts and incidents may be retained further.

There are no limitations on the size of a stored alert.

Setting the retention period for alerts and incidents

To change the retention period for alerts and incidents in the KUMA web interface:

1. In the KUMA web interface, go to the **Settings** → **Other** → **General** section.
2. In the **General** window, under **Core properties**, specify the **Alert retention period, days**.

   The retention period for alerts and incidents will be changed.

Alert and incident deletion conditions

KUMA applies the following alert and incident deletion conditions:

- If an alert is older than the retention period, its events are deleted regardless of the alert status.
- If an alert is older than the retention period, has the **Closed** status, and is not linked to an incident, the alert is deleted.
- If an alert is older than the retention period (regardless of the alert status) and is linked to an incident whose retention period has not yet expired, the alert events are deleted, the alert without events is available for display and remains linked to the incident without the events.
- If the incident is older than the retention period, the status of the incident is **Closed**, and it has no linked alerts, the incident is deleted.
- If an incident is older than the retention period, the status of the incident is **Closed**, and only alerts older than the retention period are linked to it (the status of alerts linked to a closed incident is always **Closed, incident closed**), the incident is deleted together with all its alerts and events.
- If an incident is older than the retention period (regardless of the status of the incident), and it has linked alerts that have not yet expired, nothing is deleted.
- If an incident is older than the retention period, has a status other than **Closed**, and alerts older than the retention period are linked to the incident, the incident is not deleted, alerts are not deleted, but events of the alerts are deleted.
- Empty alerts (alerts without events whose retention period has expired) are deleted together with incidents when the incident satisfes deletion conditions in accordance with the retention period are satisfied (the incident has the **Closed** status and its retention period has expired).
- If an incident is older than the retention period, the status of the incident is **Closed**, and it has linked alerts that are older than the retention period and some alerts that fall within the retention period, such an incident is not deleted. Alerts whose retention period has not expired are not deleted. Events of alerts whose retention period has expired are deleted and the alerts themselves remain empty and linked to the incident.

Alert notifications

Standard KUMA notifications are sent by email when alerts are generated and assigned. You can configure delivery of alert generation notifications based on a custom email template.

Creating a notification rule

To configure delivery of alert generation notifications based on a custom template:

1. In the KUMA web interface, open the **Settings** → **Other** → **Alerts** → **Notification rules** section.
2. Select the tenant for which you want to create a notification rule:
   - If the tenant already has notification rules, select it in the table.
   - If the tenant does not have notification rules, click **Add settings for a new tenant**.
3. This opens the **Notification rule settings** window; in that window, manage the following settings:
   - If you want the rule to be applied after saving the settings, enable the **State** toggle switch.
   - In the **Tenant** drop-down list, select a tenant.
4. Under **Notification rules**, click **Add** and specify the notification rule settings:
   - **Name** (required)—specify the notification rule name in this field.
   - **Recipient emails** (required)—in this group of settings, you can use the **Email** button to add the email addresses to which you want to send notifications about alert generation. Addresses are added one at a time.

     > Email addresses must contain only ASCII characters.

   - **Correlation rules** (required)—in this settings block, you must select one or more correlation rules that, when triggered, will cause notification sending.

     The window displays a tree structure representing the correlation rules from the shared tenant and the user-selected tenant. To select a rule, select the check box next to it. You can select the check box next to a folder to select all correlation rules in that folder and its subfolders.

   - **Template** (required)—in this settings block, you must select a notification template that will be used to create the notifications. To select a template, click the icon, select the required template in the opened window, and click **Save**.

     You can create a template by clicking the plus icon or edit the selected template by clicking the pencil icon.

   - **Disabled**—by selecting this check box, you can disable the notification rule.
5. Click **Save**.

The notification rule is created. When an alert is created based on the selected correlation rules, notifications created based on custom email templates will be sent to the specified email addresses. Standard KUMA notifications about the same event will not be sent to the specified addresses.

Duplicating the settings of a notification rule

To duplicate alert notification settings:

1. Open notification rules:

   a. In the KUMA web interface, go to the **Settings** → **Other** → **Alerts** → **Notification rules** section.

   b. In the table of tenants, select the tenant whose notification rule settings you want to duplicate. Selecting a tenant opens the **Notification rule settings** window.

2. Copy the settings of the original rule:

   a. In the **Notification rule settings** window, in the **Notification rules** section, open the rule that you need.

   This opens the **Edit notification rule** window.

   b. In the **Edit notification rule** window, click the **Copy** button.

   The **Paste** button becomes available only after you click the **Copy** button.

   You can copy within a tenant or between tenants.

   c. Click the (×) button or click **Cancel** to close the **Edit notification rule** window.

3. Preparing for pasting:

   - If you want to duplicate settings within the same tenant, proceed to the next step.
   - If you want to use the copied settings in another tenant:

     1. Close the **Notification rule settings** window for the current tenant.

     2. Navigate to your target tenant. Selecting a tenant opens the **Notification rule settings** window.

4. Pasting settings:

   a. In the **Notification rule settings** window, click **Add**.

   b. This opens the **Edit notification rule** window; in that window, click the **Paste** button.

5. Checking and editing fields. After pasting, the following settings are duplicated, and the corresponding fields are pre-filled:

   - The **State** toggle switch is set to the same position as in the original rule.
   - In the **Name** field, enter a new name for the rule. You cannot save the copy with the same name as the original rule.
   - The list of **Recipient addresses** is duplicated. You can change the port if necessary.
   - **Tenants and correlation rules**.

The behavior of the tree of correlation rules depends on whether the rule belongs to the tenant:

- If you copy and paste settings within a tenant, after pasting, the rules from the source notification rule automatically become selected.
- If you copy and paste between tenants, but the correlation rule is in the Shared tenant, after pasting, the correlation rules from the original notification rule are automatically selected in the tree of correlation rules.
- If you copy and paste between tenants, and the correlation rule is not in the Shared tenant, an access error is returned after pasting. In this case, you need to manually specify the tenant and the correlation rules.

- The same **Template** is set as in the source rule.

If necessary, edit the pasted settings.

6. Click **Save**.

The settings of the notification rule are duplicated.

Disabling a notification rule

To disable notification rules for a tenant:

1. In the KUMA web interface, open the **Settings** → **Other** → **Alerts** → **Notification rules** section and select the tenant whose notification rules you want to disable.
2. Select the **Disabled** check box.
3. Click **Save**.

The notification rules of the selected tenant are disabled.

For disabled notification rules, the correctness of the specified parameters is not checked; at the same time, notifications cannot be enabled for a tenant if incorrect rules exist. If you create or edit individual notification rules with tenant notification rules disabled, before enabling tenant notification rules, it is recommended to: 1) disable all individual notification rules, 2) enable tenant notification rules, 3) enable individual notification rules one by one.

Working with incidents

In the **Incidents** section of the KUMA web interface, you can create, view, and process incidents. You can also filter incidents if needed. Clicking the name of an incident opens a window containing information about the incident.

Incidents can be exported to NCIRCC.

The retention period for incidents is one year, but this setting can be changed.

The date format of the incident depends on the localization language selected in the solution settings. Possible date format options:

- English localization: YYYY-MM-DD.
- Russian localization: DD.MM.YYYY.

About the incidents table

The main part of the **Incidents** section shows a table containing information about registered incidents. If required, you can change the set of columns and the order in which they are displayed in the table.

How to customize the incidents table

1. Click the gear ( ) icon in the upper-right corner of the incidents table.

   The table customization window opens.

2. Select the check boxes opposite the settings that you want to view in the table.

   When you select a check box, the events table is updated and a new column is added. When a check box is cleared, the column disappears.

   You can search for table parameters using the **Search** field.

   By pressing the **Default** button, the following columns are selected for display:

   - **Name**.
   - **Threat duration**.
   - **Assigned**.
   - **Created**.
   - **Tenant**.
   - **Status**.
   - **Hits count**.
   - **Severity**.
   - **Affected asset categories**.

3. Change the display order of the columns as needed by dragging the column headings.
4. If you want to sort the incidents by a specific column, click its title and select one of the available options in the drop-down list: **Ascending** or **Descending**.
5. To filter incidents by a specific parameter, click on the column header and select the required filters from the drop-down list. The set of filters available in the drop-down list depends on the selected column.
6. To remove filters, click the relevant column heading and select **Clear filter**.

Available columns of the incidents table:

- **Name**—the name of the incident. Names of incidents received from NCIRCC have the ALRT* prefix.
- **Threat duration**—the time span during which the incident occurred (the time between the first and the last event related to the incident).
- **Assigned to**—the name of the security officer to whom the incident was assigned for investigation or response.
- **Created**—the date and time when the incident was created. This column allows you to filter incidents by the time they were created.
  - The following preset periods are available: **Today**, **Yesterday**, **This week**, **Previous week**.
  - If necessary, you can set an arbitrary period by using the calendar that opens when you select **Before date**, **After date**, **In period**.
- **Tenant**—the name of the tenant that owns the incident.
- **Status**—current status of the incident:
  - **Opened**—new incident that has not been processed yet.
  - **Assigned**—the incident has been processed and assigned to a security officer for investigation or response.
  - **Closed**—the incident is closed; the security threat has been resolved.
- **Alerts number**—the number of alerts included in the incident. Only the alerts of those tenants to which you have access are taken into account.
- **Severity** shows how important a possible security threat is: **Critical**, **High**, **Medium**, **Low**.
- **Affected asset categories**—categories of alert-related assets with the highest severity. No more than three categories are displayed.
- **Updated**—the date and time of the last change made in the incident.
- **First event** and **Last event**—dates and times of the first and last events in the incident.
- **Incident category** and **Incident type**—category and type of threat assigned to the incident.
- **Export to NCIRCC**—the status of incident data export to NCIRCC:
  - **Not exported**—the data was not forwarded to NCIRCC.
  - **Export failed**—an attempt to forward data to NCIRCC ended with an error, and the data was not transmitted.
  - **Exported**—data on the incident has been successfully transmitted to NCIRCC.
- **Branch**—data on the specific node where the incident was created. Incidents of your node are displayed by default. This column is displayed only when hierarchy mode is enabled.
- **CII**—an indication of whether the incident involves assets that are CII objects. The column is hidden from the users who do not have access to CII objects.

In the **Search** field, you can enter a regular expression for searching incidents based on their related assets, users, tenants, correlation rules, or affected asset categories. Parameters that can be used for a search:

- Assets: name, ID, FQDN, IP address, MAC address.
- Active Directory accounts: attributes displayName, SAMAccountName, and UserPrincipalName.
- Correlation rules: name.
- KUMA users who were assigned alerts: name, login, email address.
- Tenants: name. You can select one or more values from the list of tenants.
- Affected asset categories. You can select one or more categories and specify a severity.

When filtering incidents based on a specific parameter, the corresponding column in the incidents table is highlighted in yellow.

Saving and selecting incident filter configuration

In KUMA, you can save changes to incident table settings as filters. Filter configurations are saved on the KUMA Core server and are available to all KUMA users of the tenant for which they were created.

To save the current filter configuration settings:

1. In the **Incidents** section of KUMA, open the **Select filter** drop-down list.
2. Select **Save current filter**.

   A window will open for entering the name of the new filter and selecting the tenant that will own the filter.
3. Enter a name for the filter configuration. The name must be unique for alert filters, incident filters, and event filters.
4. In the **Tenant** drop-down list, select the tenant that will own the filter and click **Save**.

The filter configuration is now saved.

To select a previously saved filter configuration:

1. In the **Incidents** section of KUMA, open the **Select filter** drop-down list.
2. Select the configuration you want.

The filter configuration is now active.

You can select the default filter by putting an asterisk to the left of the required filter configuration name in the **Filters** drop-down list.

To reset the current filter settings,

open the **Filters** drop-down and select **Clear filter**.

Deleting incident filter configurations

To delete a previously saved filter configuration:

1. In the **Incidents** section of KUMA, open the **Filters** drop-down list.
2. Click the button next to the configuration you want to delete.
3. Click **OK**.

   The filter configuration is now deleted for all KUMA users.

Viewing information about an incident

To view information about an incident:

1. In the solution web interface window, select the **Incidents** section.
2. Select the incident whose information you want to view.

   This opens a window containing information about the incident.

Some incident parameters are editable. Incidents automatically created in KUMA based on a notification from NCIRCC have the ALRT* prefix.

In the upper part of the Incident details window, there is a toolbar and the name of the user to whom the incident is assigned. The window sections are displayed as tabs. You can click a tab to move to the relevant section. In this window, you can process the incident: assign it to a user, combine it with another incident, or close it.

The **Description** section contains the following data:

- **Created**—the date and time when the incident was created.
- **Name**—the name of the incident.

  You can change the name of an incident by entering a new name in the field and clicking **Save** The name must contain 1 to 128 Unicode characters.
- **Tenant**—the name of the tenant that owns the incident.

  The tenant can be changed by selecting the required tenant from the drop-down list and clicking **Save**
- **Status**—current status of the incident:
  - **Opened**—new incident that has not been processed yet.
  - **Assigned**—the incident has been processed and assigned to a security officer for investigation or response.
  - **Closed**—the incident is closed; the security threat has been resolved.

- **Severity**—the severity of the threat posed by the incident. Possible values:
  - **Critical**
  - **High**
  - **Medium**
  - **Low**

  Severity can be changed by selecting the required value from the drop-down list and clicking **Save**.
- **Affected asset categories**—the assigned categories of assets associated with the incident.
- **First event time** and **Last event time**—dates and times of the first and last events in the incident.
- **Type** and **Category**—type and category of the threat assigned to the incident. You can change these values by selecting the relevant value from the drop-down list and clicking **Save**.
- **Export to NCIRCC**—information on whether or not this incident was exported to NCIRCC.
- **Description**—description of the incident.

  To change the description, edit the text in the field and click **Save**. The description can contain no more than 256 Unicode characters.
- **Related tenants**—tenants associated with incident-related alerts, assets, and users.
- **Available tenants**—tenants whose alerts can be linked to the incident automatically.

  The list of available tenants can be changed by checking the boxes next to the required tenants in the drop-down list and clicking **Save**.

The **Related alerts** section contains a table of alerts related to the incident. When you click on the alert name, a window opens with detailed information about this alert.

The **Related endpoints** and **Related users** sections contain tables with data on assets and users related to the incident. This information comes from alerts that are related to the incident.

You can add data to the tables in the **Related alerts**, **Related endpoints** and **Related users** sections by clicking the **Link** button in the appropriate section and selecting the object to be linked to the incident in the opened window. If required, you can unlink objects from the incident. To do this, select the objects as required, click **Unlink** in the section to which they belong, and save the changes. If objects were automatically added to the incident, they cannot be unlinked until the alert mentioning those objects is unlinked. The set of the fields in the tables can be changed by clicking the gear ( ) button in the relevant section. You can search the data in the tables of these sections using the **Search** fields.

The **Change log** section contains a record of the changes you and your users made to the incident. Changes are automatically logged, but it is also possible to add comments manually.

In the **NCIRCC integration** section, you can monitor the incident status in NCIRCC. In this section, you can also export incident data to NCIRCC, send files to NCIRCC, and exchange messages with NCIRCC experts.

If incident settings have been modified on the NCIRCC side, a corresponding notification is displayed in the incident window in KUMA. In this case, for the settings whose values were modified, the window displays the values from KUMA and the values from NCIRCC.

Incident creation

To create an incident:

1. Open the KUMA web interface and select the **Incidents** section.
2. Click **Create incident**.

   The window for creating an incident will open.
3. Fill in the mandatory parameters of the incident:
   - In the **Name** field enter the name of the incident. The name must contain 1 to 128 Unicode characters.
   - In the **Tenant** drop-down list, select the tenant that owns the created incident.
4. If necessary, provide other parameters for the incident:
   - In the **Severity** drop-down list, select the severity of the incident. Available options: **Low**, **Medium**, **High**, **Critical**.
   - In the **First event time** and **Last event time** fields, specify the time range in which events related to the incident were received.
   - In the **Category** and **Type** drop-down lists, select the category and type of the incident. The available incident types depend on the selected category.
   - Add the incident **Description**. The description can contain no more than 256 Unicode characters.
   - In the **Available tenants** drop-down list, select the tenants whose alerts can be linked to the incident automatically.
   - In the **Related alerts** section, add alerts related to the incident.

     Linking alerts to incidents

     To link an alert to an incident:

     1. In the **Related alerts** section of the incident window click **Link**.

        A window with a list of alerts not linked to incidents will open.
     2. Select the required alerts.

        PCRE regular expressions can be used to search alerts by user, asset, tenant, and correlation rule.
     3. Click **Link**.

     Alerts are now related to the incident and displayed in the **Related alerts** section.

To unlink alerts from an incident:

1. Select the relevant alerts in the **Related alerts** section and click **Unlink**.
2. Click **Save**.

   Alerts have been unlinked from the incident. Also, the alert can be unlinked from the incident in the alert window using the **Unlink** button.

- In the **Related endpoints** section, add assets related to the incident.

  Linking assets to incidents

  To link an asset to an incident:

  1. In the **Related endpoints** section of the incident window, click **Link**.

     A window containing a list of assets will open.
  2. Select the relevant assets.

     You can use the **Search** field to look for assets.
  3. Click **Link**.

     Assets are now linked to the incident and are displayed in the **Related endpoints** section.

  To unlink assets from an incident:

  1. Select the relevant assets in the **Related endpoints** section and click **Unlink**.
  2. Click **Save**.

     The assets are now unlinked from the incident.

- In the **Related users** section, add users related to the incident.

  Linking users to incidents

  To link a user to an incident:

  1. In the **Related users** section of the incident window, click **Link**.

     The user list window opens.
  2. Select the required users.

     You can use the **Search** field to look for users.
  3. Click **Link**.

     Users are now linked to the incident and appear in the **Related users** section.

To unlink users from the incident:

1. Select the required users in the **Related users** section and click the **Unlink** button.
2. Click **Save**.

   Users are unlinked from the incident.

- Add a **Comment** to the incident.

5. Click **Save**.

The incident has been created.

Incident processing

For convenience of processing incidents, make sure that time is synchronized on all devices involved in the event life cycle (event sources, KUMA servers, client hosts) with the help of Network Time Protocol (NTP) servers.

You can assign an incident to a user, aggregate it with other incidents, or close it.

To process an incident:

1. Select required incidents using one of the methods below:

   - In the **Incidents** section of the KUMA web interface, click the incident to be processed.

     The incident window will open, displaying a toolbar on the top.

   - In the **Incidents** section of the KUMA web console, select the check box next to the required incidents.

     A toolbar will appear at the bottom of the window.

2. In the **Assign to** drop-down list, select the user to whom you want to assign the incident.

   You can assign the incident to yourself by selecting **Me**.

   The status of the incident changes to **assigned** and the name of the selected user is displayed in the **Assign to** drop-down list.

3. In the **Related users** section, select a user and configure Active Directory response settings.

   a. After the related user is selected, in the **Account details** window that opens, click **Response via Active Directory**.

   b. In the **AD command** drop-down list, select one of the following 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

- Block account

c. Click **Apply**.

4. If required, edit the incident parameters.

5. After investigating, close the incident:

   a. Click **Close**.

   A confirmation window opens.

   b. Select the reason for closing the incident:

   - **confirmed**. This means the incident was valid and appropriate measures were taken to eliminate the security threat.

   - **not confirmed**. This means the incident was a false positive and the received events do not indicate a security threat.

   c. Click **Close**.

   The **Closed** status will be assigned to the incident. Incidents with this status cannot be edited, and they are displayed in the incidents table only if you selected the **Closed** check box in the **Status** drop-down list when filtering the table. You cannot change the status of a closed incident or assign it to another user, but you can aggregate it with another incident.

6. If requited, aggregate the selected incidents with another incident:

   a. Click **Merge**. In the opened window, select the incident in which all data from the selected incidents should be placed.

   b. Confirm your selection by clicking **Merge**.

   The incidents will be aggregated.

The incident has been processed.

Changing incidents

To change the parameters of an incident:

1. In the **Incidents** section of the KUMA web interface, click on the incident you want to modify.

   The Incident window opens.

2. Make the necessary changes to the parameters. All incident parameters that can be set when creating it are available for editing.

3. Click **Save**.

The incident will be modified.

Automatic linking of alerts to incidents

In KUMA, you can configure automatic linking of generated alerts to existing incidents if alerts and incidents have related assets or users in common. If this setting is enabled, when creating an alert the solution searches for incidents falling into a specified time interval that includes assets or users from the alert. In addition, the solution checks whether the generated alert pertains to the tenants specified in the incidents' **Available tenants** parameter. If a matching incident is found, the solution links the generated alert to the incident it found.

To set up automatic linking of alerts to incidents:

1. In the KUMA web interface, open **Settings** → **Incidents** → **Automatic linking of alerts to incidents**.

2. Select the **Enable** check box in the **Link by assets** and/or **Link by accounts** groups of settings depending on the types of connections between incidents and alerts that you are looking for.

3. Define the **Incidents must not be older than** value for the parameters that you want to use when searching links. The generated alerts will be compared with incidents no older than the specified interval.

Automatic linking of alerts to incidents is configured.

To disable automatic linking of alerts to incidents,

In the KUMA web interface, under **Settings** → **Incidents** → **Automatic linking of alerts to incidents**, select the **Disabled** check box.

Categories and types of incidents

For your convenience, you can assign categories and types. If an incident has been assigned a NCIRCC category, it can be exported to NCIRCC.

Categories and types of incidents that can be exported to NCIRCC

The table below lists the categories and types of incidents that can be exported to NCIRCC:

Incident category	Incident type
Computer incident notification	Slowed operation of the resource due to a DDoS attack
	Malware infection
	Network traffic interception
	Compromised user account
	Unauthorized data modification
	Unauthorized disclosure of information
	Publication of illegal information on the resource
	Successful exploitation of a vulnerability
	Event is not related to a computer attack
	Use of a controlled resource for attacks
Notification about a computer attack	DDoS attack
	Unsuccessful authorization attempts
	Malware injection attempts
	Attempts to exploit a vulnerability
	Publication of fraudulent information
	Network scanning
	Social engineering
Notification about a detected vulnerability	Vulnerable resource

The categories of incidents can be viewed or changed under **Settings** → **Incidents** → **Incident types**, in which they are displayed as a table. By clicking on the column headers, you can change the table sorting options. The resource table contains the following columns:

- **Category**—a common characteristic of an incident or cyberattack. The table can be filtered by the values in this column.
- **Type**—the class of the incident or cyberattack.
- **NCIRCC category**—incident type according to NCIRCC nomenclature. Incidents that have been assigned custom types and categories cannot be exported to NCIRCC. The table can be filtered by the values in this column.
- **Vulnerability**—specifies whether the incident type indicates a vulnerability.
- **Created**—the date the incident type was created.
- **Updated**—the date the incident type was modified.

To add an incident type:

1. In the KUMA web interface, under **Settings** → **Incidents** → **Incident types**, click **Add**.

   The incident type creation window will open.

2. Fill in the **Type** and **Category** fields.
3. If the created incident type matches the NCIRCC nomenclature, select the **NCIRCC category** check box.

4. If the incident type indicates a vulnerability, check **Vulnerability**.

5. Click **Save**.

The incident type has been created.

Interaction with NCIRCC

In KUMA, you can interact with the National Computer Incident Response & Coordination Center (hereinafter NCIRCC) in the following ways:

- Export incidents to NCIRCC.
- Supplement the exported incident with data when requested by NCIRCC.
- Send files to NCIRCC.
- Exchange messages with NCIRCC experts.
- View the changes made by NCIRCC to the parameters of exported incidents.
- Process incidents with the ALRT* prefix that are automatically created in KUMA based on a notification from NCIRCC.

Data in KUMA and NCIRCC is synchronized every 5-10 minutes.

Conditions for NCIRCC interaction

To interact with NCIRCC, the following conditions must be met:

- The solution license includes the GosSOPKA module.
- NCIRCC integration is configured.
- The **Can interact with NCIRCC** check box is selected in the settings of the users whose responsibilities include interaction with NCIRCC.

NCIRCC interaction workflow

In KUMA, the process of sending incidents to NCIRCC to be processed consists of the following stages:

1. **Creating an incident and checking it for compliance with NCIRCC requirements**

   You can create an incident or get it from a child KUMA node. Before sending data to the NCIRCC, make sure that the incident category meets NCIRCC requirements.

2. **Exporting the incident to NCIRCC**

If the incident is successfully exported to NCIRCC, its **Export to NCIRCC** setting is set to **Exported**. In the lower part of the incident window, a chat with NCIRCC experts becomes available.

At NCIRCC, the incident received from you is assigned a registration number and status. This information is displayed in the incident window in the **NCIRCC integration** section and in automatic chat messages.

If all the necessary data is provided to NCIRCC, the incident is assigned the **Under examination** status. The settings of the incident having this status can be edited, but the updated information cannot be sent from KUMA to NCIRCC. You can view the difference between the incident data in KUMA and in NCIRCC.

3. **Supplementing incident data**

   If NCIRCC experts do not have enough information to process an incident, they can assign it the **More information required** status. In KUMA, this status is displayed in the incident window in the **NCIRCC integration** section. Users are notified about the status change.

   If necessary, you can attach a file to the incident.

   When the data is supplemented, the incident is re-exported to NCIRCC with earlier information updated. The incidents in the child nodes cannot be modified from the parent KUMA node. It must be done by employees of the child KUMA nodes.

   If the incident is successfully supplemented with data, it is assigned the **Under examination** status.

4. **Completing incident processing**

   After the NCIRCC experts process the incident, the NCIRCC status is changed to **Decision made**. In KUMA, this status is displayed in the incident window in the **NCIRCC integration** section.

   Upon receiving this status, the incident is automatically closed in KUMA. Interaction with NCIRCC on this incident by means of KUMA becomes impossible.

Exporting data to RuCERT

To export an incident to NCIRCC:

1. In the **Incidents** section of the KUMA web interface, open the incident you want to export.
2. Click the **Export to NCIRCC** button in the lower part of the window.
3. If you have not specified the category and type of incident, specify this information in the window that opens and click the **Export to NCIRCC** button.

   This opens the export settings window.
   If you specified the category and type of the incident in the incident card, you need to save the incident before exporting it to NCIRCC.
4. Specify the settings on the **Basic** tab of the **Export to NCIRCC** window:
5. On the **Basic settings** tab, fill in the required fields:

   Company name, **Asset owner**, Incident category, Incident type, **Description**, **TLP protocol value**, **Incident creation date**, **Status**, **Affected system name**, CII system category, **Business line**, **Location**.

   - Nuclear energy
   - Banking and other financial market sectors
   - Mining
   - Federal/municipal government

- Healthcare
- Metallurgy
- Science
- Defense industry
- Education
- Aerospace industry
- Communication
- Mass media
- Fuel and power
- Transportation
- Chemical industry
- Other
- WHITE—disclosure is not restricted.
- GREEN—disclosure is only for the community.
- AMBER—disclosure is only for organizations.
- RED—disclosure is only for a specific group of people.
- If you want to submit information about a personal data leak, select the **Information about PD leakage** check box, which enables the **Information about PD leakage** tab. This check box is cleared by default.

  The **Operator name**, **INN**, and **Operator address** fields are populated automatically with the values specified when setting up the NCIRCC integration. However, the field values can be edited when preparing the data export to NCIRCC.

  Personal data leak fields are available for the **Notification about a computer incident** category for the following notification types:

  - **Malware infection**
  - **Compromised user account**
  - **Unauthorized disclosure of information**
  - **Successful exploitation of a vulnerability**
  - **Event is not related to a computer attack**
- **Product info** (required)—this table becomes available if you selected **Notification about a detected vulnerability** as the incident category.

  You can use the **Add new element** button to add a string to the table. In the **Name** column, you must indicate the name of the application (for example, `MS Office`). Specify the application version in the **Version** column (for example, `2.4`).

- **Vulnerability ID**—if necessary, specify the identifier of the detected vulnerability. For example, `CVE-2020-1231`.

  This field becomes available if you selected **Notification about a detected vulnerability** as the incident category.

- **Product category**—if necessary, specify the name and version of the vulnerable product. For example, `Microsoft operating systems and their components`.

  This field becomes available if you selected **Notification about a detected vulnerability** as the incident category.

6. If required, define the settings on the **Advanced** tab of the **Export to NCIRCC** window.

   The available settings on the tab depend on the selected category and type of incident:

   - **Detection tool**—specify the name of the product that was used to register the incident. For example, `KUMA 1.5`.
   - **Assistance required**—select this check box if you need help from GosSOPKA employees.
   - **Incident end time**—specify the date and time when the critical information infrastructure (CII object) was restored to normal operation after a computer incident, computer attack was ended, or a vulnerability was fixed.
   - **Availability impact**—assess the degree of impact that the incident had on system availability:
     - **High**
     - **Low**
     - **None**
   - **Integrity impact**—assess the degree of impact that the incident had on system integrity:
     - **High**
     - **Low**
     - **None**
   - **Confidentiality impact**—assess the degree of impact that the incident had on data confidentiality:
     - **High**
     - **Low**
     - **None**
   - **Custom impact**—specify other significant impacts from the incident.
   - **City**—indicate the city where your organization is located.

7. If assets are attached to the incident, you can specify their settings on the **Technical details** tab.

   This tab becomes active only if you select the **Affected system has internet connection** check box.

If you need to edit or supplement the information previously specified on the **Technical details** tab, you should do this in your GosSOPKA account, even if NCIRCC experts requested additional information from you, and you can edit the exported incident.

The categories of the listed assets must match the category of the affected CII in your system.

8. Click **Export**.

9. Confirm the export.

   Information about the incident is submitted to NCIRCC, and the **Export to NCIRCC** incident setting is changed to **Exported**. At NCIRCC, the incident received from you is assigned a registration number and status. This information is displayed in the incident window in the **NCIRCC integration** section.

It is possible to change the data in the exported incident only if the NCIRCC experts requested additional information from you. If no additional information was requested, but you need to update the exported incident, you should do it in your GosSOPKA dashboard.

After the incident is successfully exported, the **Compare KUMA incident to NCIRCC data** button is displayed at the bottom of the screen. When you click this button, a window opens, where the differences in the incident data between KUMA and NCIRCC are highlighted.

Getting incidents from NCIRCC

After upgrading to version 3.4.1, at each startup, KUMA Core sends a request to get new incident cards to the address specified in the URL field in KUMA's NCIRCC integration settings, and then continues sending requests every 10 minutes. If a new incident appears in the NCIRCC user account dashboard, KUMA registers the incident with the ALRT* prefix and further interaction with NCIRCC is carried out in context of the created incident.

Interaction with NCIRCC is available even if the incident in KUMA has the **Closed** status: you can edit the value of the **NCIRCC status** field and chat with NCIRCC.

Notifications about messages from NCIRCC are sent to the email addresses of users with the General administrator role.

Fields of a new incident are populated as shown in the following table.

Field values of an incident received from NCIRCC.

Field name	Value
Created by	When an incident is created in KUMA, creation date and time are automatically specified. Example: 2024-10-08 05:32:39
Name	ID or registration number of the incident (message) in GosSOPKA. Example: ALRT-xx-xx-xxx
Tenant	The tenant in which incidents received from NCIRCC must be created. By default, all incidents are created in the Main tenant. You can change the incident creation tenant in the NCIRCC integration settings. Example: Main.
Status	Initial status of the KUMA incident. Example: **Open**.
Severity	Degree of significance of the potential security threat. Default: **Critical**. Possible values: **Critical**, **High**, **Medium**, **Low**.
Affected asset categories	Empty. This value can be filled in manually.
First event time	Empty. This value can be filled in manually.

Field name	Value
Last event time	Empty. This value can be filled in manually.
Description	The value from the event_description field of the message sent by NCIRCC. Can be edited. Example: message from NCIRCC.
Related tenants	Same as the Tenant field. Example: Main.
Available tenants	Same as the Tenant field. Example: Main.
Related alerts	Empty. This value can be filled in manually.
Related endpoints	Empty. This value can be filled in manually.
Related users	Empty. This value can be filled in manually.
Change log	Empty. This value can be filled in manually.
NCIRCC integration section	
Category	Category of the message card. Dictionary value. Possible values: Message from NCIRCC. This field is not editable.
Type	Type of the information security event. Value from the dictionary. Possible values: **Infected resource**, **Source of malware email messages**, **Source of malware modules**, **Malware operation center**, **Malware infrastructure element**, **Resource slowdown**, **Source of exploit**, **Source of user account compromise**, **Participant of network traffic capture**, **Source of unauthorized access**, **Source of unauthorized modification of information**, **Source of spam**, **Publication of information illegal in the Russian Federation**, **Hosting of a phishing resource**, **Unauthorized content**, **Participant of DDoS attack**, **Compromised user account**, **Network Scan Source**, **Participant of fraud**, **Source of social engineering threat**, **Vulnerable resource**, **Suspected phishing resource**, **Threat of personal data compromise**, **Threat of computer attack**. This field is not editable.
Status of export to NCIRCC	**Imported from NCIRCC**. This field is not editable.
NCIRCC message processing progress section	
NCIRCC status	NCIRCC message processing status. Value from the dictionary. This field is editable. Possible values: **New**, **In progress**, **Decision made**, **Interaction completed**, **Archived**.
Result	Result of the decision made. Value from the dictionary. This field is editable. Possible values: **Measures have been taken**, **Information taken into account**, **Information not confirmed**.
Chat with NCIRCC section	
UUID	Unique ID of the NCIRCC message.
Company	Short name of the GosSOPKA subject organization.
Category	Category of the message card. Dictionary value. Example: message from NCIRCC.
Type	Type of the information security event. Value from the dictionary. Example: infected resource.
Creation time	Date and time when the NCIRCC message was registered. Follows the ISO 8601 standard. UTC time is used.
Time of detection	Date and time when the incident was detected. Follows the ISO 8601 standard. UTC time is used.
Incident end time	Date and time when the incident ended. Follows the ISO 8601 standard. UTC time is used.
Updated	Date and time when the message card was last updated. Follows the ISO 8601 standard. UTC time is used.
Event description	Brief description of the information security event. Up to 5000 Unicode characters.
Owner name	Owner of the information resource. Up to 5000 Unicode characters.
Incident ID in NCIRCC	Registration number of the message. Example: ALRT-20-12-2914.
TLP	TLP limitation marker. Dictionary value. Possible values: **TLP:WHITE**, **TLP:GREEN**, **TLP:AMBER**, **TLP:RED**.

Field name	Value
Technical details section	
The **Technical details** group of fields is displayed with the field values obtained from the message. Which **NCIRCC message** fields are present in the **Technical details** group depends on the Information security event type, and the set of fields may differ depending on the incident. Pay close attention to the field values in your NCIRCC account. Values are not editable.	

Supplementing incident data on request

If NCIRCC experts need additional information about the incident, they may request it from you. In this case, the incident status changes to **More information required** in the **NCIRCC integration** section of the incident window. The following KUMA users receive email notifications about the status change: the user to whom the incident is assigned and the user who exported the incident to NCIRCC.

If an incident is assigned the *More information required* status in NCIRCC, the following actions are also available for this incident in KUMA:

- Upload files to NCIRCC.
- Re-export the incident data to NCIRCC with updates or additions to the previously provided information. This action completes supplementing the incident data.

Sending files to NCIRCC

You can attach a file to an incident if the incident has one of the following NCIRCC statuses: *New*, *In progress*, *More information required*, *Decision made*. The file will be available both in NCIRCC and in the KUMA web interface.

For a hierarchical deployment of KUMA, files can only be uploaded to NCIRCC from the parent KUMA node. At the same time, log entries about the file download are visible in the child nodes of KUMA.

In the incident change log, messages about the files uploaded to NCIRCC by KUMA users are added. Messages about adding files by NCIRCC are not added to the log.

To attach a file to an incident:

1. In the **Incidents** section of the KUMA web interface, open the incident you want to attach a file to.
2. In the **NCIRCC integration** section of the incident window, select the **File** tab and click the **Send file to NCIRCC** button.

   The file selection window opens.
3. Select the required file no larger than 50 MB and confirm your selection.

The file is attached to the incident and available for both NCIRCC experts and KUMA users.

Data in KUMA and NCIRCC is synchronized every 5-10 minutes.

Communication with NCIRCC experts

After the incident is successfully exported to NCIRCC, a chat with NCIRCC experts becomes available at the bottom of the screen. You can exchange messages since successful incident export to NCIRCC until it is closed in NCIRCC.

The chat window with the message history and the field for entering new messages is available on the **Chat** tab in the **NCIRCC integration** section of the incident window.

Data in KUMA and NCIRCC is synchronized every 5-10 minutes.

Supported categories and types of NCIRCC incidents

The table below lists the categories and types of incidents that can be exported to NCIRCC:

Incident category	Incident type
Computer incident notification	Slowed operation of the resource due to a DDoS attack
	Malware infection
	Network traffic interception
	Compromised user account
	Unauthorized data modification
	Unauthorized disclosure of information
	Publication of illegal information on the resource
	Successful exploitation of a vulnerability
	Event is not related to a computer attack
	Use of a controlled resource for attacks
Notification about a computer attack	DDoS attack
	Unsuccessful authorization attempts
	Malware injection attempts
	Attempts to exploit a vulnerability
	Publication of fraudulent information
	Network scanning
	Social engineering
Notification about a detected vulnerability	Vulnerable resource

Notifications about the incident status change in NCIRCC

In the event of certain changes in the status or data of an incident at NCIRCC, KUMA users receive the following notifications by email:

- Notification about receiving a message from NCIRCC.
- Additional data request notification.

- Notification about the incident status change in NCIRCC.
- Notification about automatic closure of an incident.

The following users receive notifications:

- The user to whom the incident was assigned.
- The user who exported the incident to NCIRCC.

Retroscan

In normal mode, the correlator handles only events coming from collectors in real time. **Retrospective scan** allows you to apply correlation rules to past events if you want to debug correlation rules or analyze data.

To test a correlation rule, you do not need to replay the incident in real time. Instead, you can run the correlation rule in **Retrospective scan** mode to process the existing events which include the required incident.

You can use a search query to define a list of events to retrospectively scan, you can also specify a search period and the storage that you want to search for events. You can configure a task to have alerts generated and response rules applied during the retroscan of events.

To use Retroscan:

1. In the **Events** section of the KUMA web interface, create the required event selection:
   - Select the storage.

     Before running the retroscan, make sure that the necessary clusters and spaces are selected.
   - Configure search expression using the constructor or search query.

     Before running the retroscan, make sure that the correct expression is specified in the SQL query field.
   - Set a period as needed.

     Before running the retroscan, make sure that the correct period is selected in the period field.
2. Open the ••• drop-down list and choose **Retrospective scan**.
   The Retroscan window opens.

3. In the **Correlator** drop-down list, select the Correlator to feed selected events to.

   Before running the retroscan, check the number of events found by the SQL query. This information is displayed under **Show timeline** → **Total**. If the **LIMIT** parameter value is not specified in the SQL query, all events for the period are sent to the correlator and the task takes longer to complete. If a relative period is set (for example, *now-1h*), new events may be produced by the start of the retrospective scan; such events are also sent to the correlator. If the correlator is not installed in the Main tenant, it can only receive events of its own tenant.

4. In the **Correlation rules** drop-down list, select the Correlation rules that must be used when processing events. If no rules are selected at this step, the scan is performed with all correlation rules applied.

5. If you want responses to be executed when processing events, turn on the **Execute responses** toggle switch.

6. If you want alerts to be generated during event processing, turn on the **Create alerts** toggle switch.

7. Click the **Create task** button.

   The retroscan task is created and displayed in the **Task manager** section.

To view scan results, in the **Task manager** section of the KUMA web interface, click the task you created and in the drop-down list, select **Go to Events**.

Retroscun results only show the generated correlation events.
The start of the period in retroscan results is displayed as <date and time of the task creation>, and the end of the period as <date and time of task creation + one hour> (or the current time if one hour has not yet elapsed since the task creation).
If there are no events in the retroscan results, it means that the correlator was not triggered by events from the SQL selection that were sent to it.

A new browser tab is opened, containing events that were processed during the retroscan and the aggregation and correlation events that were created during event processing. Correlation events generated by the retroscan have an additional ReplayID field that stores the unique ID of the retrospective scan run. An analyst can restart the retroscan from the context menu of the task. New correlation events will have a different ReplayID.

Depending on your browser settings, you may be prompted to confirm the opening of the retroscan results tab. For more details, please refer to the documentation for your specific browser.

Retroscanned events are not enriched with data from CyberTrace or the Kaspersky Threat Intelligence Portal.

Active lists are updated during retrospective scanning.

Retroscan cannot be performed based on SQL queries that contain arithmetic expressions or only aggregation fields. You can retroscan groups of events obtained using a grouping SQL query only after the processing of the query has finished. To do this, first run your grouping SQL query and get a list of event groups, and then run retroscan on one of the groups.

Contacting Technical Support

If you did not find a solution to your problem in the documentation or other sources of information about KUMA, we recommend contacting Technical Support. Our Technical Support team will answer your questions about installing and using KUMA.

Kaspersky provides support for KUMA throughout its lifecycle (see the solution lifecycle page). Before contacting Technical Support, please read the technical support rules.

You can contact Technical Support in one of the following ways:

- By visiting the Technical Support website
- By submitting a request to Kaspersky Technical Support from the Kaspersky CompanyAccount portal.

When processing your request, Kaspersky Technical Support may need a report generated by the collect.sh utility. For detailed information on using the utility, please refer to the Knowledge Base.

REST API

You can access KUMA from third-party solutions using the API. The KUMA REST API operates over HTTP and consists of a set of request/response methods. REST API v3 is supported.

REST API requests must be sent to the following address:

https://<KUMA Core FQDN>:<REST API port>/api/<API version>/<request>

Example:
https://kuma.example.com:7223/api/v3

By default the 7223 port is used for API requests. You can change the port.

To change port used for REST API requests:

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-core.service change the following string, adding required port:

   ```
   ExecStart=/opt/kaspersky/kuma/kuma core --external :7220 --internal :7210 --rest
   <required port number for REST API requests>
   ```

3. Restart KUMA by running the following commands in sequence:

   a. `systemctl daemon-reload`

   b. `systemctl restart kuma-core`

   New port is used for REST API.

Make sure that the port is available and is not closed by the firewall.

Authentication header: Authorization: Bearer <token>

Default data format: JSON

Date and time format: RFC 3339

Intensity of requests: unlimited

Creating a token

To generate a token for a user:

1. In the KUMA web interface, open **Settings** → **Users**.

   In the right part of the **Settings** section the **Users** table will be displayed.

2. Select the relevant user and click the **Generate token** button in the details area that opens on the right.

   The **New token** window opens.

3. 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.

4. Click the **Generate token** button.

   When you click this button, the user details area displays a field containing the automatically created token. 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.

5. Click **Save**.

The token is generated and can be used for API requests. These same steps can be taken to generate a token in your account profile.

Configuring permissions to access the API

In KUMA, you can configure which operations each user can perform. Permissions can be configured only for users created in KUMA.

To configure available operations for a user:

1. In the KUMA web interface, open **Settings** → **Users**.

   In the right part of the **Settings** section the **Users** table will be displayed.

2. Select the relevant user and click the **API access rights** button in the details area on the right.

   This opens a window containing a list of available operations. By default, all API requests are available to a user.

3. Select or clear the check box next to the relevant operation.

4. Click **Save**.

Available operations for the user are configured.

The available operations can be configured in the same way in your account profile.

Authorizing API requests

Each REST 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	details field value
400	Invalid header	invalid authorization header	Example: <example>
403	The token does not exist or the owner user is disabled	access denied	

Standard error

Errors returned by KUMA have the following format:

```
type Error struct {
    Message    string      `json:"message"`
    Details    interface{} `json:"details"`
}
```

REST API v3 operations

Open REST API Help

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 a 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 the version of the solution.

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

You can check the integrity of KUMA components manually by running one of the scripts described below.

The integrity of KUMA components is checked using a set of scripts based on the integrity_checker tool and located in the/opt/kaspersky/kuma/integrity/bin directory. An integrity check uses manifest XML files from the /opt/kaspersky/kuma/integrity/manifest/* directory, protected by a Kaspersky cryptographic signature.

Running the integrity check tool requires a user account with permissions at least matching those of the KUMA account.

The integrity check tool processes each KUMA component individually, and it must be run on servers that has the appropriate components installed. An integrity check also checks the xml file that was used.

To check the integrity of component files:

1. Run the following command to navigate to the directory that contains the set of scripts:

   ```
   cd /opt/kaspersky/kuma/integrity/bin
   ```

2. Then run one of the following commands depending on the KUMA component that you want to check:

   - `./check_all.sh` for KUMA Core and Storage components.
   - `./check_core.sh` for the KUMA Core component.
   - `./check_collector.sh` for the KUMA collector component.
   - `./check_storage.sh` for the storage component.
   - `./check_event_router.sh` for the event router component.
   - `./check_kuma_exe.sh <full path to kuma.exe omitting file name>` for KUMA Agent for Windows. The standard location of the agent executable file on the Windows device is: C:\Program Files\Kaspersky Lab\KUMA\
   - `./check_linux_agent.sh <full path to the executable file of the KUMA agent without the file name>` for the KUMA agent for Linux.

   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 – the number of KUMA files processed.
  - Directories – not used when KUMA integrity checking is performed.
  - Registries – not used when KUMA integrity checking is performed.
  - Registry values – not used when KUMA integrity checking is performed.

- Component integrity check result:
  - SUCCEEDED – integrity confirmed.
  - FAILED – integrity violated.

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. The maximum size of an event that can be processed by the KUMA collector is 4 MB.

Events that are not compliant to this data model must be converted to this format (or normalized) using Collectors.

Normalized event data model

Field name	Data type	Field size	Description
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, Telnet.
`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 on. For example, 0.0.0.0 or xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
`DestinationCity`	String	1,023 characters	City corresponding to the IP address from the DestinationAddress field.
`DestinationCountry`	String	1023 characters	Country corresponding to the IP address from the DestinationAddress field.
`DestinationDnsDomain`	String	255 characters	The DNS portion of the fully qualified domain name of the destination.
`DestinationHostName`	String	1023 characters	Host name of the destination. FQDN of the destination, if available.
`DestinationLatitude`	Floating point number	+/- 1.7E-308 to 1.7E+308	Longitude corresponding to the IP address from the DestinationAddress field.
`DestinationLongitude`	Floating point number	+/- 1.7E-308 to 1.7E+308	Latitude corresponding to the IP address from the DestinationAddress field.
`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 example, sshd, telnet.

DestinationRegion	String	1023 characters	Region corresponding to the IP address from the DestinationAddress field.
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, 0.0.0.0 or xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
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 example, "User", "Guest", or "Administrator".
DeviceAction	String	63 characters	Action that was taken by the event source. For example, blocked, detected.
DeviceAddress	String	45 characters	IPv4 or IPv6 address of the device from which the event was received. For example, 0.0.0.0 or xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
DeviceCity	String	1023 characters	City corresponding to the IP address from the DeviceAddress field.
DeviceCountry	String	1023 characters	Country corresponding to the IP address from the DeviceAddress field.
DeviceDnsDomain	String	255 characters	DNS part of the fully qualified domain name of the device from which the event was received.
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. If you want the DeviceExternalID field to be displayed as a link in the event properties, configure a template in the enrichment rule or in the correlation rule.
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 the device, if available.
DeviceInboundinterface	String	128 characters	Name of the incoming connection interface.
DeviceLatitude	Floating point number	+/- 1.7E-308 to 1.7E+308	Longitude corresponding to the IP address from the DeviceAddress field.
DeviceLongitude	Floating point number	+/- 1.7E-308 to 1.7E+308	Latitude corresponding to the IP address from the DeviceAddress field
DeviceMacAddress	String	17 characters	MAC address of the asset from which the event was received. For example, aa:bb:cc:dd:ee:00
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 DeviceVendor, DeviceProduct, and DeviceVersion all uniquely 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 field.
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 event was received. For example, 0.0.0.0 or xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
DeviceVendor	String	63 characters	Vendor name of the event source. The DeviceVendor, DeviceProduct, and DeviceVersion all uniquely identify the log source.
DeviceVersion	String	31 characters	Application version of the event source. The DeviceVendor, DeviceProduct, and DeviceVersion all uniquely identify the log 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.
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 specified in UTC0. In the KUMA web interface, the value is displayed based in the timezone of the user's browser.
OldFileHash	String	255 characters	Hash of the OLD file.
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	2,048 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	Priority. This can be the Severity field or the Level field of the raw event.
SourceAddress	String	45 characters	IPv4 or IPv6 address of the source. Example format: 0.0.0.0 or xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
SourceCity	String	1023 characters	City corresponding to the IP address from the SourceAddress field.
SourceCountry	String	1023 characters	Country corresponding to the IP address from the SourceAddress field.
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	Floating point number	+/- 1.7E-308 to 1.7E+308	Longitude corresponding to the IP address from the SourceAddress field.
SourceLongitude	Floating point number	+/- 1.7E-308 to 1.7E+308	Latitude corresponding to the IP address from the SourceAddress field.
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, telnet.
SourceRegion	String	1023 characters	Region corresponding to the IP address from the SourceAddress field.
SourceServiceName	String	1023 characters	Name of the service on the source side. For example, sshd.
SourceTranslatedAddress	String	45 characters	Translated IPv4 or IPv6 address of the source. Example format: 0.0.0.0 or xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
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 example, "User", "Guest", or "Administrator".
StartTime	Number	From -9223372036854775808 to 9223372036854775807	Date and time (timestamp) when the activity associated with the event began.
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 OSI Transport layer protocol (for example, TCP, UDP).
Type	Number	From -9223372036854775808 to 9223372036854775807	Event type: 1 - basic, 2 - aggregated, 3 - correlation, 4 - audit, 5 - monitoring.
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 specified in UTC0. In the KUMA web interface, the value is 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 specified in UTC0. In the KUMA web interface, the value is displayed based in the timezone of the user's browser.
DeviceCustomDate2Label	String	1023 characters	Field for describing the purpose of the DeviceCustomDate2 field.
DeviceCustomFloatingPoint1	Floating point number	+/- 1.7E-308 to 1.7E+308	Field for mapping floating point numbers.
DeviceCustomFloatingPoint1Label	String	1023 characters	Field for describing the purpose of the DeviceCustomFloatingPoint1 field.
DeviceCustomFloatingPoint2	Floating point number	+/- 1.7E-308 to 1.7E+308	Field for mapping floating point numbers.
DeviceCustomFloatingPoint2Label	String	1023 characters	Field for describing the purpose of the DeviceCustomFloatingPoint2 field.
DeviceCustomFloatingPoint3	Floating point number	+/- 1.7E-308 to 1.7E+308	Field for mapping floating point numbers.
DeviceCustomFloatingPoint3Label	String	1023 characters	Field for describing the purpose of the DeviceCustomFloatingPoint3 field.
DeviceCustomFloatingPoint4	Floating point number	+/- 1.7E-308 to 1.7E+308	Field for mapping floating point numbers.
DeviceCustomFloatingPoint4Label	String	1023 characters	Field for describing the purpose of the DeviceCustomFloatingPoint4 field.
DeviceCustomIPv6Address1	String	45 characters	Field for mapping an IPv6 address value. Format example: y:y:y:y:y:y:y:y
DeviceCustomIPv6Address1Label	String	1023 characters	Field for describing the purpose of the DeviceCustomIPv6Address1 field.
DeviceCustomIPv6Address2	String	45 characters	Field for mapping an IPv6 address value. Format example: y:y:y:y:y:y:y:y
DeviceCustomIPv6Address2Label	String	1023 characters	Field for describing the purpose of the DeviceCustomIPv6Address2 field.
DeviceCustomIPv6Address3	String	45 characters	Field for mapping an IPv6 address value. Format example: y:y:y:y:y:y:y:y
DeviceCustomIPv6Address3Label	String	1023 characters	Field for describing the purpose of the DeviceCustomIPv6Address3 field.
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 DeviceCustomIPv6Address4 field.
DeviceCustomNumber1	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer value.

DeviceCustomNumber1Label	String	1023 characters	Field for describing the purpose of the DeviceCustomNumber1 field.
DeviceCustomNumber2	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer value.
DeviceCustomNumber2Label	String	1023 characters	Field for describing the purpose of the DeviceCustomNumber2 field.
DeviceCustomNumber3	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer value.
DeviceCustomNumber3Label	String	1023 characters	Field for describing the purpose of the DeviceCustomNumber3 field.
DeviceCustomString1	String	4,000 characters	Field for mapping a string value.
DeviceCustomString1Label	String	1,023 characters	Field for describing the purpose of the DeviceCustomString1 field.
DeviceCustomString2	String	4,000 characters	Field for mapping a string value.
DeviceCustomString2Label	String	1023 characters	Field for describing the purpose of the DeviceCustomString2 field.
DeviceCustomString3	String	4,000 characters	Field for mapping a string value.
DeviceCustomString3Label	String	1023 characters	Field for describing the purpose of the DeviceCustomString3 field.
DeviceCustomString4	String	4,000 characters	Field for mapping a string value.
DeviceCustomString4Label	String	1023 characters	Field for describing the purpose of the DeviceCustomString4 field.
DeviceCustomString5	String	4,000 characters	Field for mapping a string value.
DeviceCustomString5Label	String	1023 characters	Field for describing the purpose of the DeviceCustomString5 field.
DeviceCustomString6	String	4,000 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" - incoming connection, "1" - outgoing connection.
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 specified in UTC0. In the KUMA web interface, the value is 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 and user accounts, and find out the number of times they appear in alert events.
AggregationRuleID	String	-	Aggregation rule ID.
AggregationRuleName	String	-	The name of the aggregation rule that processed the event.
BaseEventCount	Number	-	For an aggregated base event, this is the number of base events that were processed by the aggregation rule. For a correlation event, this is the number of base events that were processed by the correlation rule that generated the correlation event.
BaseEvents	Nested [Event] list	-	Nested structure containing a list of base events. This field can be filled in for correlation events.
Code	String	-	In a base event, this is the code of a process, function or operation return from the source.
CorrelationRuleID	String	-	ID of the correlation rule.
CorrelationRuleName	String	-	Name of the correlation rule that triggered the creation of the correlation event. Filled only for correlation events.
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 those fields that have not been mapped to KUMA event fields. This field can be filled in only for base events. The maximum size of the field is 4 MB.
GroupedBy	String	-	List of names of the fields that were used for grouping in the correlation rule. It is filled in only for the correlation event.
ID	String	-	Unique event ID of UUID type. For a base event that is generated on the collector, the ID is generated by the collector. The correlator generates the ID of a correlation event. The ID never changes its value.
Raw	String	-	Non-normalized text of the original raw event. Maximum field size is 16,384 bytes.
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 assigns when creating the microservice.
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 an external Threat Intelligence source based on indicators from an event.
`Timestamp`	Number	-	Timestamp of the base event created in the collector. Creation time of the correlation event created by the collector. The time is specified in UTC0. In the KUMA web interface, the value is displayed based in the timezone of the user's browser.

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	How many 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, TenantID (the field displays the name of the tenant, an enriched value, while the tenant ID is used for searching the database).

Configuring the data model of a normalized event from KATA EDR

When investigating incidents, it can be useful to build a chain of processes. For this purpose, it is important that certain fields of the normalized event contain KATA/EDR process IDs. After correctly configuring normalization, you can build a process chain for events coming from KATA/EDR using an SQL query.

Example of building a process chain using an SQL query

Example of an SQL query for selecting events containing a process chain

```
SELECT
```

```
    b.DeviceHostName AS HostName,
    b.SourceAccountID AS AccountID,
    b.SourceUserName AS UserName,
    a.DeviceCustomString3 AS GrandParentProcessID,
    a.SourceProcessName AS GrandParentProcessName,
    a.DeviceCustomString5 AS ParentProcessID,
    a.DestinationProcessName AS ParentProcessName,
    b.DeviceCustomString5 AS ProcessID,
    b.DestinationProcessName AS ProcessName,
    concat(a.SourceProcessName, ' -> ', a.DestinationProcessName, ' -> ',
b.DestinationProcessName) AS ProcessTree
FROM `events` AS a
INNER JOIN
    (
        SELECT *
        FROM `events`
        WHERE DeviceEventClassID='4688'
    ) AS b
ON a.DeviceCustomString5 = b.DeviceCustomString3
WHERE DeviceEventClassID='4688'
```

Result

As a result of the query execution, events that meet the query conditions are returned in the **Events** section. In the event card, the `ProcessTree` field lists the process chain: process A → process B → … → process N.

For example:

```
C:\Windows\System32\svchost.exe -> C:\Windows\WinSxS\amd64_microsoft-windows-
servicingstack_31bf3856ad364e35_10.0.17763.164_none_7e114a3d4d0589d4\TIWorker.exe ->
C:\Windows\System32\conhost.exe
```

To build a process chain, you need to 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

KATA/EDR event field	Normalized event field
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
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

Event	Raw event field	Normalized event field
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
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

Event	Raw event field	Normalized event field
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

Alert data model

This section describes the KUMA alert data model. Alerts are created by correlators whenever information security threats are detected using correlation rules. Alerts must be investigated to eliminate these threats.

Alert field	Data type	Description
`ID`	String	Unique ID of the alert.
`TenantID`	String	ID of the tenant that owns the alert. The value is inherited from the correlator that generated the alert.
`TenantName`	String	Tenant name.
`CorrelationRuleID`	String	ID of the rule used as the basis for generating the alert.
`CorrelationRuleName`	String	Name of the correlation rule used as the basis for generating the alert.
`Status`	String	Alert status. Possible values: • `New`—new alert. • `Assigned`—the alert is assigned to a user. • `Closed`—the alert was closed. • `Exported to IRP`—the alert was exported to the IRP system for further investigation. • `Escalated`—an incident was generated based on this alert.
`Priority`	Number	Alert severity. Possible values: • 1–4 — Low. • 5–8 — Medium. • 9–12 — High. • 13–16 — Critical.
`ManualPriority`	`TRUE/FALSE` string	Parameter showing how the alert severity level was determined. Possible values: • `true`—defined by the user. • `false` (default value)—calculated automatically.
`FirstSeen`	Number	Time when the first correlation event was created from the alert.
`LastSeen`	Number	Time when the last correlation event was created from the alert.
`UpdatedAt`	Number	Date of the last modification to the alert parameters.

`UserID`	String	ID of the KUMA user assigned to examine the alert.
`UserName`	String	Name of the KUMA user assigned to examine the alert.
`GroupedBy`	Nested list of strings	List of event fields used to group events in the correlation rule.
`ClosingReason`	String	Reason for closing the alert. Possible values: • `Incorrect Correlation Rule` —the alert was a false positive and the received events do not indicate a real security threat. The correlation rule may need to be updated. • `Incorrect Data` —the alert was a false positive and the received events do not indicate a real security threat. • `Responded` —the appropriate measures were taken to eliminate the security threat.
`MaxAssetsWeightStr`	String	Maximum severity of the asset categories associated with the alert.
`MaxAssetCategoryImportance`	String	This field contains the value of the `MaxAssetsWeightStr` field in a format suitable for display. Possible values: • `Unknown` • `Low` • `Medium` • `High` • `Critical`
`IntegrationID`	String	ID of the alert in the IRP / SOAR application, if integration with such an application is configured in KUMA.
`ExternalReference`	String	Link to a section in the IRP / SOAR application that displays information about an alert imported from KUMA.
`IncidentID`	String	ID of the incident to which the alert is linked.
`IncidentName`	String	Name of the incident to which the alert is linked.
`SegmentationRuleName`	String	Name of the segmentation rule used to group correlation events in the alert.
`BranchID`	String	ID of the hierarchy branch in which the alert was generated. Indicated for a hierarchical deployment of KUMA.
`BranchName`	String	Name of the hierarchy branch in which the alert was generated. Indicated for a hierarchical deployment of KUMA.
`Actions`	Nested `[Action]` structure	Nested structure with lines indicating changes to alert statuses and assignments, and user comments.
`Events`	Nested `[EventWrapper]` structure	Nested structure from which you can query the correlation events associated with the alert.
`Assets`	Nested `[Asset]` structure	Nested structure from which you can query assets associated with the alert.
`Accounts`	Nested `[Account]` structure	Nested structure from which you can query the user accounts associated with the alert.
`AffectedAssets`	Nested `[Affected]` structure	Nested structure from which you can query alert-related assets and user accounts, and find out the number of times they appear in alert events.

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.

Nested `EventWrapper` structure

Field	Data type	Description
`Event`	Nested `[Event]` structure	Event fields.
`Comment`	String	Comment added when events were added to the alert.
`LinkedAt`	Number	Date when events were added to the alert.

Nested `Action` structure

Field	Data type	Description
`CreatedAt`	Number	Date when the action was taken on the alert.
`UserID`	String	User ID.
`Kind`	String	Type of action.
`Value`	String	Value.
`Event`	Nested `[Event]` structure	Event fields.
`ClusterID`	String	Cluster ID.

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/KATA server IP address.
`KICSConnectorID`	Number	KICS/KATA connector ID.
`KICSDeviceID`	Number	KICS/KATA asset ID.
`KICSStatus`	String	KICS/KATA asset status.
`KICSHardware`	Nested `[KICSSystemInfo]` structure	Asset hardware information received from KICS/KATA.
`KICSSoft`	Nested `[KICSSystemInfo]` structure	Asset software information received from KICS/KATA.
`KICSRisks`	Nested `[KICSRisk]` structure	Asset vulnerability information received from KICS/KATA.
`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 KSC.
`NAgentID`	String	ID of the KSC Agent from which the asset information was received.
`KSCServerFQDN`	String	FQDN of the KSC Server.
`KSCInstanceID`	String	KSC instance ID.
`KSCMasterHostname`	String	KSC Server host name.
`KSCGroupID`	Number	KSC group ID.
`KSCGroupName`	String	KSC group name.
`LastVisible`	Number	Date when information about the asset was last received from KSC.
`Products`	Nested dictionary: `[string:nested [ProductInfo] structure]`	Information about Kaspersky applications installed on the asset received from KSC.
`Hardware`	Nested `[Hardware]` structure	Asset hardware information received from KSC.
`KSCSoft`	Nested `[Software]` structure	Asset software information received from KSC.
`KSCVulns`	Nested `[Vulnerability]` structure	Asset vulnerability information received from KSC.

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`	`TRUE/FALSE` string	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`	`TRUE/FALSE` string	Indicates whether an exploit exists.
`MalwareExists`	`TRUE/FALSE` string	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/KATA 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 KSC.
`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/KATA.

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 structure `ProductInfo`

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. AD 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. AD 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.

KUMA 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.

User was successfully signed in or failed to sign in

Event field name	Field value
`DeviceAction`	`user login`
`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.
SourceUserID	User ID.
Message	Description of the error. Appears only if an error occurred during login. Otherwise, the field will be empty.

User login changed

Event field name	Field value
DeviceAction	user login changed
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 data.
SourceUserID	ID of the user that modified the data.
DestinationUserName	User login whose data was changed.
DestinationUserID	ID of the user whose data was changed.
DeviceCustomString1	Current value of the login.
DeviceCustomString1Label	new login
DeviceCustomString2	Value of the login before it was changed.
DeviceCustomString2Label	old login

Other data of the user was changed

Event field name	Field value
DeviceAction	user other info changed
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 data.
SourceUserID	ID of the user that modified the data.
DestinationUserName	User login whose data was changed.
DestinationUserID	ID of the user whose data was changed.

User logged out

This event appears only when the user pressed the logout button.

This event will not appear if the user is logged out due to the end of the session or if the user logs in again from another browser.

Field values of audit event User logged out

Event field name	Field value
DeviceAction	user logout
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.
SourceUserID	User ID.

User password was changed

Event field name	Field value
DeviceAction	user password changed
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 data.
SourceUserID	ID of the user that modified the data.
DestinationUserName	User login whose data was changed.
DestinationUserID	ID of the user whose data was changed.

User was created

Event field name	Field value
DeviceAction	user 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 user account.
SourceUserID	ID of the user that created the user account.
DestinationUserName	User login for which the user account was created.
DestinationUserID	User ID for which the user account was created.
DeviceCustomString1	Role of the created user.
DeviceCustomString1Label	role

User role was assigned

Event field name	Field value
DeviceAction	granted access
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 whose data was changed.
SourceUserID	ID of the user whose data was changed.
DestinationUserPrivileges	Role ID of the user whose data was changed.
DeviceCustomString5	ID of the tenant used to assign the role.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

User role was revoked

Event field name	Field value
DeviceAction	revoked access
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	Login of the user that made the changes.
SourceUserID	ID of the user that made the changes.
DestinationUserName	User login whose data was changed.
DestinationUserID	ID of the user whose data was changed.
DestinationUserPrivileges	Role ID of the user whose data was changed.
DeviceCustomString5	ID of the tenant used to assign the role.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

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.

Alert assigned to the user

Event field name	Field value
DeviceAction	user is assigned to alert
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 assigned the alert to the user.
SourceUserID	ID of the user that assigned the alert to the user.
ExternalID	Alert ID.
Name	Alert name.
DestinationUserName	User name of the user to which the alert was assigned.
DestinationUserID	ID of the user to which the alert was assigned.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID

DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

User access token was changed

Event field name	Field value
DeviceAction	user access token changed
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 data.
SourceUserID	ID of the user that modified the data.
DestinationUserName	User login whose data was changed.
DestinationUserID	ID of the user whose data was changed.

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.

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.

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

License renewed

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.

Activation code blocked

Event field name	Field value
DeviceAction	Activation code 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.

Activation code or key file deleted

Event field name	Field value
DeviceAction	Activation code or key file 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.

Activation code or key file replaced

Event field name	Field value
DeviceAction	Activation code or key file replaced
EventOutcome	succeeded
Message	License name.
SourceUserID	ID of the user that performed the replacement.
SourceUserName	User name of the user that performed the replacement.
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 activation code or key file added

Event field name	Field value
DeviceAction	Activation code or key file 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

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
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
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
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 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 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 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. The error text can be in json format, for example, {"error":"line 1: header does not contain column IP!"}.
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.

Asset was created

Event field name	Field value
DeviceAction	asset 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 login that was used to add the asset.
SourceUserID	User ID that was used to add the asset.
DeviceAssetID	Asset ID.
SourceHostName	Asset ID.
Name	Asset name.
DeviceCustomString1	Comma-separated IP addresses of the asset.
DeviceCustomString1Label	addresses

DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Asset was deleted

Event field name	Field value
DeviceAction	asset 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 add the asset.
SourceUserID	User ID that was used to add the asset.
DeviceAssetID	Asset ID.
SourceHostName	Asset ID.
Name	Asset name.
DeviceCustomString1	Comma-separated IP addresses of the asset.
DeviceCustomString1Label	addresses
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Asset category was added

Event field name	Field value
DeviceAction	category 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 login that was used to add the category.
SourceUserID	User ID that was used to add the category.
DeviceExternalID	Category ID.

Name	Category name.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Asset category was deleted

Event field name	Field value
DeviceAction	category 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 category.
SourceUserID	User ID that was used to delete the category.
DeviceExternalID	Category ID.
Name	Category name.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

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

Tenant was created

Event field name	Field value
DeviceAction	tenant 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 login used to create the tenant.
SourceUserID	User ID used to create the tenant.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Tenant was enabled

Event field name	Field value
DeviceAction	tenant enabled
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 login used to enable the tenant.
SourceUserID	User ID used to enable the tenant.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Tenant was disabled

Event field name	Field value
DeviceAction	tenant disabled
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 login used to disable the tenant.
SourceUserID	User ID used to disable the tenant.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Other tenant data was changed

Event field name	Field value
DeviceAction	tenant other info changed
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 login that was used to change the tenant data.
SourceUserID	User ID that was used to change the tenant data.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

The dictionary was successfully updated on the service or operation was unsuccessful

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

Incident successfully created or operation failed

Event field name	Field value
DeviceAction	incident created
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 created the incident.
SourceUserID	ID of the user that created the incident.
ExternalID	ID of the incident.
Name	Name of the incident.
DeviceCustomString1	Alert name. This field is filled if the incident was created based of an alert. If multiple alerts were involved, these are specified as a comma-separated list.
DeviceCustomString1Label	alert name
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Incident successfully closed or operation failed

Event field name	Field value
DeviceAction	incident closed
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 closed the incident.
SourceUserID	ID of the user that closed the incident.
ExternalID	ID of the incident.
Name	Name of the incident.
DeviceCustomString1	Decision: confirmed or not confirmed.
DeviceCustomString1Label	resolution
Message	If EventOutcome = failed, an error message is written to this field.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Incident assigned to a user or operation failed

Event field name	Field value
DeviceAction	user is assigned to incident
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 assigned the user.
SourceUserID	ID of the user that assigned the user.
ExternalID	ID of the incident.
Name	Name of the incident.
DestinationUserName	User name of the user to which the incident was assigned.
DestinationUserID	ID of the user to which the incident was assigned.
Message	If EventOutcome = failed, an error message is written to this field.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Alert linked to an incident or alert unlinked from an incident, or operation failed

Event field name	Field value
DeviceAction	link alert to incident or unlink alert from incident
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 linked the alert to the incident or unlinked the alert from the incident.
SourceUserID	ID of the user that linked the alert to the incident or unlinked the alert from the incident.
ExternalID	ID of the incident.
Name	Name of the incident.
DeviceFacility	Action: link or unlink.
DeviceCustomString1	ID of the alert that was linked or unlinked.
DeviceCustomString1Label	alert ID
DeviceCustomString2	Name of the alert that was linked or unlinked.
DeviceCustomString2Label	alert name
Message	If EventOutcome = failed, an error message is written to this field.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

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

Event linked to an alert or unlinked from an alert, or operation failed

Event field name	Field value
DeviceAction	link event to alert or unlink event from alert
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 linked or unlinked the event.
SourceUserID	ID of the user that linked or unlinked the event.
ExternalID	Alert ID.
Name	Alert name.
DeviceFacility	Event action: link or unlink.
DeviceCustomString1	ID of the event that was linked or unlinked.

DeviceCustomString1Label	event ID
Message	If EventOutcome = failed, an error message can be found here.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

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

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.

Backing up KUMA Core and restoring from backup

Event field name	Field value
DeviceAction	backup or restore
EventOutcome	succeeded or failed
SourceUserName	User name of the user that performed the backup or restoration from backup.
SourceUserID	ID of the user that performed the backup or restoration from backup.
DeviceExternalID	ID of the KUMA Core service.
Name	Name of the KUMA Core service.
FilePath	Full path at which the backup file is saved.
Message	If EventOutcome = failed, an error message can be found here.
DeviceCustomString5	ID of the tenant that owns the KUMA Core service.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Backing up partitions, restoring from backup, importing partitions from an external KUMA installation

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.

Correlation rules

The file that can be downloaded by clicking the link describes the correlation rules that are included in the distribution kit of Kaspersky Unified Monitoring and Analysis Platform version 4.2. It provides the scenarios covered by rules, the conditions of their use, and the necessary sources of events.

Only one of the following versions of the SOC rule set can be used at a time:

- [OOTB] SOC Content - RU or [OOTB] SOC Content - ENG
- [OOTB] SOC Content - RU for KUMA 3.2 or [OOTB] SOC Content - ENG for KUMA 3.2
- [OOTB] KSC Package - RU or [OOTB] KSC Package - ENG
- [OOTB] KSMG Package - RU or [OOTB] KSMG Package - ENG
- [OOTB] KATA package - RU or [OOTB] KATA package - ENG
- [OOTB] Network Package - RU or [OOTB] Network Package - ENG
- [OOTB] PCIDSS Package - RU or [OOTB] PCIDSS Package - ENG
- [OOTB] XDR package - RU or [OOTB] XDR package - ENG
- [OOTB] UEBA package - RU or [OOTB] UEBA package - ENG
- [OOTB] PostgreSQL package - RU or [OOTB] PostgreSQL package - ENG

You can import correlation rules into KUMA. See the *Importing resources* section of the Help: https://support.kaspersky.com/KUMA/4.2/en-US/242787.htm.

You can add imported correlation rules to correlators that your organization uses. See the Help section *Step 3. Correlation*: https://support.kaspersky.com/KUMA/4.2/en-US/221168.htm.

Download the description of KSC Package correlation rules

Download the description of KSMG Package correlation rules

Download the description of KATA Package correlation rules

Download the description of KUMA Package correlation rules

Download the description of Network Package correlation rules

Download the description of PCIDSS Package correlation rules

Download a description of SOC Content for KUMA correlation rules

Download the description of UEBA Package correlation rules

Download the description of XDR Package correlation rules

Download the description of PostgreSQL Package correlation rules

Description of correlation rule packages

The distribution kit of Kaspersky Unified Monitoring and Analysis Platform 4.2 includes the correlation rule packages listed in the table below.

Package name	Description
[OOTB] SOC Content - RU	Correlation rule package for KUMA version 2.1 or later with Russian localization. This package is no longer supported.
[OOTB] SOC Content - ENG	Correlation rule package for KUMA version 2.1 or later with English localization. This package is no longer supported.
[OOTB] SOC Content - RU for KUMA 3.2	Correlation rule package for KUMA version 3.2 or later with Russian localization. The rules contain information about MITRE ATT&CK matrix coverage.
[OOTB] SOC Content - ENG for KUMA 3.2	Correlation rule package for KUMA version 3.2 or later with English localization. The rules contain information about MITRE ATT&CK matrix coverage.
[OOTB] KUMA package - RU	Correlation rules package for monitoring KUMA events. The correlation rules allow detecting potentially dangerous actions in the system based on internal audit events. The package is available for KUMA version 3.2 or later with Russian localization. Solution audit events are used to detect anomalies in user behavior, such as failed login attempts, connections from non-typical IP addresses, and password brute-force attempts. Attempts to change solution settings, including manipulation of accounts and other parameters, are also monitored. Such actions may indicate unauthorized access or bypassing security mechanisms to conduct further attacks on the corporate infrastructure. This package also includes rules for monitoring the health of services. For the rules to work correctly, make sure your audit settings provide for a sufficient level of detail in event logs.
[OOTB] KUMA package - ENG	Correlation rules package for monitoring KUMA events. The correlation rules allow detecting potentially dangerous actions in the system based on internal audit events. The package is available for KUMA version 3.2 or later with English localization. Solution audit events are used to detect anomalies in user behavior, such as failed login attempts, connections from non-typical IP addresses, and password brute-force attempts. Attempts to change solution settings, including manipulation of accounts and other parameters, are also monitored. Such actions may indicate unauthorized access or bypassing security mechanisms to conduct further attacks on the corporate infrastructure. This package also includes rules for monitoring the health of services. For the rules to work correctly, make sure your audit settings provide for a sufficient level of detail in event logs.
[OOTB] KSC Package - RU	Package of correlation rules for monitoring Kaspersky Security Center events. The package is available for KUMA version 3.2 or later with Russian localization. Allows detecting potentially dangerous actions in the system based on internal audit events. For correlation rules to work properly, make sure your audit settings provide for a sufficient level of detail in event logs. The correlation rules package works on events that are collected via the Kaspersky Security Center database. The rules contain information about MITRE ATT&CK matrix coverage.
[OOTB] KSC Package - ENG	Package of correlation rules for monitoring Kaspersky Security Center events. The package is available for KUMA version 3.2 or later with English localization. Allows detecting potentially dangerous actions in the system based on internal audit events. For correlation rules to work properly, make sure your audit settings provide for a sufficient level of detail in event logs. The correlation rules package works on events that are collected via the Kaspersky Security Center database. The rules contain information about MITRE ATT&CK matrix coverage.
[OOTB] KSMG Package - RU	Package of correlation rules for monitoring Kaspersky Security Mail Gateway events. The package is available for KUMA version 3.2 or later with Russian localization. Allows detecting potentially dangerous actions in the mail traffic protection system system based on internal audit events. For the rules to work properly, make sure your audit settings provide for a sufficient level of detail in event logs. The audit settings of Kaspersky Secure Mail Gateway are located in the **Settings** → **Logs and events** → **Events** section. In the **Audit events** section, select **Audit log level** → **Log audit events and modified parameters**. Notes: • The package of rules works on events in CEF format. • To export events to the SIEM system, you need the Kaspersky Secure Mail Gateway log in CEF (local2) format. The rules contain information about MITRE ATT&CK matrix coverage.
[OOTB] KSMG Package - ENG	Package of correlation rules for monitoring Kaspersky Security Mail Gateway events. The package is available for KUMA version 3.2 or later with English localization. Allows detecting potentially dangerous actions in the mail traffic protection system system based on internal audit events. For the rules to work properly, make sure your audit settings provide for a sufficient level of detail in event logs. The audit settings of Kaspersky Secure Mail Gateway are located in the **Settings** → **Logs and events** → **Events** section. In the **Audit events** section, select **Audit log level** → **Log audit events and modified parameters**. Notes: • The package of rules works on events in CEF format. • To export events to the SIEM system, you need the Kaspersky Secure Mail Gateway log in CEF (local2) format. The rules contain information about MITRE ATT&CK matrix coverage.

Package name	Description
[OOTB] KATA package - RU	The correlation rules package for monitoring Kaspersky Anti Targeted Attack (KATA) events allows detecting potentially dangerous activity in the system based on internal audit events. The package is available for KUMA version 3.2 or later with Russian localization. Application audit events are used to detect anomalies in user behavior, such as failed login attempts, connections from non-typical IP addresses, and password brute-force attempts. Attempts to change application settings, including manipulation of accounts and disabling of detection rules, are also monitored. Such actions may indicate unauthorized access or bypassing security mechanisms to conduct further attacks on the corporate infrastructure. For the rules to work correctly, make sure your audit settings provide for a sufficient level of detail in event logs.
[OOTB] KATA package - ENG	The correlation rules package for monitoring Kaspersky Anti Targeted Attack (KATA) events allows detecting potentially dangerous activity in the system based on internal audit events. The package is available for KUMA version 3.2 or later with English localization. Application audit events are used to detect anomalies in user behavior, such as failed login attempts, connections from non-typical IP addresses, and password brute-force attempts. Attempts to change application settings, including manipulation of accounts and disabling of detection rules, are also monitored. Such actions may indicate unauthorized access or bypassing security mechanisms to conduct further attacks on the corporate infrastructure. For the rules to work correctly, make sure your audit settings provide for a sufficient level of detail in event logs.
[OOTB] Network Package - RU	Package of correlation rules aimed at detecting network activity anomalies, for KUMA version 3.2 and later with Russian localization. The rules contain information about MITRE ATT&CK matrix coverage. The package also contains dashboards.
[OOTB] Network Package - ENG	Package of correlation rules aimed at detecting network activity anomalies, for KUMA version 3.2 and later with English localization. The rules contain information about MITRE ATT&CK matrix coverage. The package also contains dashboards.
[OOTB] PCIDSS Package - RU	Package of correlation rules aimed at compliance with the PCI DSS standard. The package is available for KUMA version 3.2 or later with Russian localization. This package also contains dashboards. For the resource set to work correctly, you need to add the **S.Compliance_PCIDSS_Requirements** extended event schema field in the **Settings** → **Other** → **Extended event schema fields**. The rules contain information about MITRE ATT&CK matrix coverage.
[OOTB] PCIDSS Package - ENG	Package of correlation rules aimed at compliance with the PCI DSS standard. The package is available for KUMA version 3.2 or later with English localization. This package also contains dashboards. For the resource set to work correctly, you need to add the **S.Compliance_PCIDSS_Requirements** extended event schema field in the **Settings** → **Other** → **Extended event schema fields**. The rules contain information about MITRE ATT&CK matrix coverage.
[OOTB] XDR package - RU	This package contains a set of resources for detecting advanced attacks in the infrastructure. The package is available for KUMA version 3.2 or later with Russian localization. It can help detect phishing (which is, according to Kaspersky MDR, the most popular vector of initial access), but also virus infection, defense evasion, payload delivery and attempts to bypass protective equipment, delivery and launch, persistence, reconnaissance, lateral movement, and various methods of communication with a command and control server. The rules can be applied to events from Kaspersky software (EDR, KSC, KSMG) as well as events from Windows operating systems. The rules contain information about MITRE ATT&CK matrix coverage.
[OOTB] XDR package - ENG	This package contains a set of resources for detecting advanced attacks in the infrastructure. The package is available for KUMA version 3.2 or later with English localization. It can help detect phishing (which is, according to Kaspersky MDR, the most popular vector of initial access), but also virus infection, defense evasion, payload delivery and attempts to bypass protective equipment, delivery and launch, persistence, reconnaissance, lateral movement, and various methods of communication with a command and control server. The rules can be applied to events from Kaspersky software (EDR, KSC, KSMG) as well as events from Windows operating systems. The rules contain information about MITRE ATT&CK matrix coverage.
[OOTB] UEBA package - RU	Beta version of the correlation rule package. This correlation rule package aims to detect abnormal behavior of an account or workstation on a corporate network. The package is available for KUMA version 3.2 or later with Russian localization. It builds adaptive models of normal behavior based on accumulated historical data (the data collection period can be configured, one month being the default). These models are used to calculate statistical indicators that minimize false positives and effectively identify potentially dangerous deviations. Sources of data: Windows OS, NetFlow, Cisco, Solar Proxy events. Notes: • Collection rules can be adapted to specific information security tools and telemetry sources. • Event enrichment with DNS data is critical to maximizing the effectiveness of the package. • Rules may affect the performance of the correlator. • We recommend using these rules in KUMA version 4.0 and higher. The rules contain information about MITRE ATT&CK matrix coverage.

Package name	Description
[OOTB] UEBA package - ENG	Beta version of the correlation rule package. This correlation rule package aims to detect abnormal behavior of an account or workstation on a corporate network. The package is available for KUMA version 3.2 or later with English localization. It builds adaptive models of normal behavior based on accumulated historical data (the data collection period can be configured, one month being the default). These models are used to calculate statistical indicators that minimize false positives and effectively identify potentially dangerous deviations. Sources of data: Windows OS, NetFlow, Cisco, Solar Proxy events. Notes: • Collection rules can be adapted to specific information security tools and telemetry sources. • Event enrichment with DNS data is critical to maximizing the effectiveness of the package. • Rules may affect the performance of the correlator. • We recommend using these rules in KUMA version 4.0 and higher. The rules contain information about MITRE ATT&CK matrix coverage.
[OOTB] NCCCI package - RU for KUMA 4.0	This package includes a set of dashboards and report templates that visualize detailed information about violations classified by categories and types of security incidents from the National Computer Incident Response and Coordination Center (NCIRCC). The visualization is based on the values of the S.Compliance_NCCCI_IncidentClassification field, which is automatically populated by an enrichment rule for rules from the SOC package. Before importing the visualization package, you must create the S.Compliance_NCCCI_IncidentClassification extended event schema field of the string type or check to make sure that it exists. This package also contains dashboards.
[OOTB] FSTEC package - RU for KUMA 4.0	This package includes a set of dashboards and report templates that visualize detailed information about violations of protective measures recommended by the Russian Federal Service for Technical and Export Control (FSTEC) to prevent threats and eliminate vulnerabilities in accordance with the Information Security Threat Database. The visualization is based on the values of the S.Compliance_FSTEC_ProtectiveMeasures field, which is automatically populated by an enrichment rule for rules from the SOC package. Before importing the visualization package, you must create the S.Compliance_FSTEC_ProtectiveMeasures extended event schema field of the string type or check to make sure that it exists. This package also contains dashboards.
[OOTB] PostgreSQL package - RU [OOTB] PostgreSQL package - ENG	Correlation rules package for monitoring PosgreSQL events. The correlation rules allow detecting potentially dangerous actions in the system based on internal audit events. Application audit events detect anomalies in user behavior, such as the gathering of information about user accounts, attempts to execute arbitrary code on the system, as well as writing and reading data from the file system. For correlation rules to work correctly, make sure your audit settings provide for a sufficient level of detail in event logs.

Automatic rule suppression

The SOC_package correlation rules package allows automatically suppressing the triggering of rules if the triggering frequency exceeds thresholds.

The automatic suppression option works as follows: if a rule is triggered more than 100 times in 1 minute and this behavior occurs at least 5 times in the span of 10 minutes, the rule is added to the stop list.

- When placed in the stop list for the first time, the rule is disabled for 1 hour.
- If this happens again, it is placed in the list for 24 hours.
- All subsequent occurrences place it in the list for 7 days.

The logic is described in the resources: rules, active lists, and dictionaries, which are located in the "SOC_package/System/Rule disabling by condition" directory.

You can customize settings and thresholds in accordance with your requirements.

To enable the automatic suppression option, set the **enable** setting to `1` in the "SOC_package/Integration/Rule disabling configuration" dictionary.

To disable the automatic suppression option, set the **enable** setting to `0` in the "SOC_package/Integration/Rule disabling configuration" dictionary.

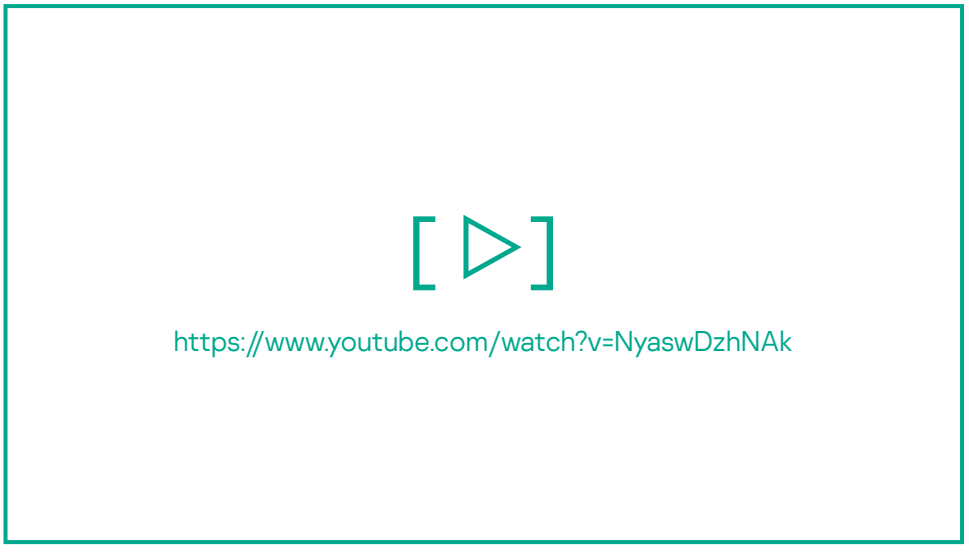
By default, automatic suppression is enabled and the **enable** setting is set to `1`.

Audit events

Correlation rules from the "[OOTB] SOC Content" resource set use the audit events that are listed in the table below.

Audit events

Event source	Audit events
CheckPoint	Anti Malware, Threat Emulation
Cisco ASA, FTD, PIX	106021, 320001, 322001, 322002, 322003, 405001, 405002
CyberTrace	alert
DNS	query
KATA	TAA has tripped on events database
KSC	GNRL_EV_ATTACK_DETECTED, GNRL_EV_SUSPICIOUS_OBJECT_FOUND, GNRL_EV_VIRUS_FOUND, GNRL_EV_WEB_URL_BLOCKED, KLSRV_HOST_STATUS_CRITICAL, KLSRV_HOST_STATUS_OK, KLSRV_HOST_STATUS_WARNING
KSMG	LMS_EV_SCAN_LOGIC_AV_STATUS, LMS_EV_SCAN_LOGIC_KT_STATUS, LMS_EV_SCAN_LOGIC_CF_STATUS, LMS_EV_SCAN_LOGIC_AP_STATUS
KUMA	Correlation rule
Windows Event Log Powershell	4103, 4104
Windows Event Log Security	1102, 4624, 4625, 4656, 4657, 4662, 4663, 4672, 4688, 4697, 4720, 4722, 4723, 4724, 4725, 4726, 4727, 4728, 4729, 4730, 4731, 4732, 4733, 4734, 4735, 4737, 4738, 4768, 4769, 4771, 5136, 5140, 5145
Windows Event Log System	7036, 7045
Windows Event Log Defender	1006, 1015, 1116, 1117, 5001, 5010, 5012, 5101
Netflow, FW	Traffic log
Palo Alto	virus
auditd	ADD_USER, DEL_USER, PATH, SYSCALL, USER_AUTH, USER_LOGIN, execve



https://www.youtube.com/watch?v=NyaswDzhNAk



https://www.youtube.com/watch?v=G2F0I7SZG6Q

Sending test events to KUMA

KUMA allows sending test events to the system. Use the option of sending test events to KUMA to test rules, reports, dashboards, and also to check the resource consumption of the collector with different event streams. Events can only be sent to a collector that receives events over TCP or HTTP.

To send test events, you need:

- The 'kuma' file started with certain options.

  In the following instructions, the file with raw events is named send_test_events.txt as an example. You can use your own file name.

- A configuration file in which you define the parameters for running the executable file.

  In the following instructions, the configuration file is named config_for_test_events as an example. You can use your own file name.

To send test events:

1. Get sample events to send to KUMA:

   a. In the KUMA web interface, in the **Events** section, in the upper right corner, click the ⚙ icon and in the displayed window, on the **Event fields columns** tab, select the check box for the **Raw** field. The 'Raw' column is displayed in the **Events** window.

   b. Search for events.

   c. Export your search results: in the **Events** window, in the upper right corner, click ... and select **Export TSV**.

   d. Go to the **Task manager** section of KUMA and click the **Events export** task; in the context menu, select **Download**.

   The `<name of file with exported events>`.tsv file is displayed in the Downloads section.

   If you are not collecting raw events, enable collection for a short time by setting the **Keep raw event** setting of the normalizer to **Always**. After the collection is completed, restore the previous value of the **Keep raw event** setting.

e. Create a text file named send_test_events.txt and copy the contents of the `Raw` field from `<name of file with exported events>`.tsv to a text file named send_test_events.txt.

f. Save send_test_events.txt.

2. Create a config_for_test_events configuration file and add the following lines to the file:

```
{
"kind": "<tcp or http>",
"name": "-",
"connection": {
"name": "-",
"kind": "<tcp or http>",
"urls": ["< IP address of the KUMA collector for receiving events over TCP >:< port of the KUMA collector for receiving event over TCP >"]
}
}
```

Save the config_for_test_events configuration file.

3. Ensure that network connectivity exists between the server sending events and the server on which the collector is installed.

4. To send the contents of the test event file to the KUMA collector, run the following command:

```
/opt/kaspersky/kuma/kuma tools load --raw --events /home/events/send_test_events.txt --cfg home/events/config_for_test_events --limit 1500 --replay 100000
```

Available settings

Setting	Description
`--events`	Full path to the file containing raw events. Required setting. If the full path is not specified, the command does not run.
`--cfg`	Path to the configuration file. Required setting. If the full path is not specified, the command does not run.
`--limit`	Stream to be sent to the collector, in events per second (EPS). Required setting. If no value is specified, the command does not run.
`--replay`	Number of events to send. Required setting. If no value is specified, the command does not run. The step for --replay is 10000. The minimum value is 10000. --replay 16 sends 10000 events. --replay 16000 sends 20000 events.

As a result of running the command, test events are successfully sent to the KUMA collector. You can verify the arrival of test events by searching for related events in the KUMA web interface.

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".
- "13842163671111222333".

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.

Deprecated resources

List of deprecated resources

Name	Resource type	Description
[Deprecated][OOTB] Microsoft SQL Server xml	Normalizer	This normalizer was removed from the resource set in KUMA 3.2. If you were using this normalizer, you must migrate to the [OOTB] Microsoft Products for KUMA 3 normalizer.
[Deprecated][OOTB] Windows Basic	Normalizer	This normalizer was removed from the resource set in KUMA 3.2. If you were using this normalizer, you must migrate to the [OOTB] Microsoft Products for KUMA 3 normalizer.
[Deprecated][OOTB] Windows Extended v.0.3	Normalizer	This normalizer was removed from the resource set in KUMA 3.2. If you were using this normalizer, you must migrate to the [OOTB] Microsoft Products for KUMA 3 normalizer.
[Deprecated][OOTB] Cisco ASA Extended v 0.1	Normalizer	This normalizer was removed from the resource set in KUMA 3.2. If you were using this normalizer, you must migrate to the [OOTB] Cisco ASA and IOS syslog normalizer.
[Deprecated][OOTB] Cisco Basic	Normalizer	This normalizer was removed from the resource set in KUMA 3.2. If you were using this normalizer, you must migrate to the [OOTB] Cisco ASA and IOS syslog normalizer.
[Deprecated][OOTB] Linux audit and iptables syslog	Normalizer	This normalizer was removed from the resource set in KUMA 3.4. In KUMA 3.2, we recommend using the [OOTB] Linux auditd syslog for KUMA 3.2 normalizer.
[Deprecated][OOTB] Linux audit.log file	Normalizer	This normalizer was removed from the resource set in KUMA 3.4. In KUMA 3.2, we recommend using the [OOTB] Linux auditd file for KUMA 3.2 normalizer.
[OOTB] Checkpoint Syslog CEF by CheckPoint	Normalizer	This normalizer was removed from the resource set in KUMA 3.4. In KUMA 3.4 and newer versions, we recommend using the [OOTB] Checkpoint syslog normalizer.
[OOTB] Eltex MES Switches	Normalizer	This normalizer was removed from the resource set in KUMA 3.4. In KUMA 3.4 and newer versions, we recommend using the [OOTB] Eltex MES syslog, [OOTB] Eltex ESR syslog normalizers.

Name	Resource type	Description
[OOTB] PTsecurity NAD	Normalizer	This normalizer was removed from the resource set in KUMA 3.4. In KUMA 3.4 and newer versions, we recommend using the [OOTB] PTsecurity NAD json normalizer.
[OOTB][AD] Granted TGS without TGT (Golden Ticket)	Rule	The rule was removed from the KUMA 3.4 resource set.
[OOTB][AD] Possible Kerberoasting attack	Rule	The rule was removed from the KUMA 3.4 resource set.
[OOTB][AD][Technical] 4768. TGT Requested	Rule	The rule was removed from the KUMA 3.4 resource set.
[OOTB][AD] List of requested TGT. EventID 4768	Active list	The list was removed from the KUMA 3.4 resource set.
[OOTB] Kaspersky Industrial CyberSecurity for Networks 4.2 syslog	Normalizer	This normalizer is obsolete and was removed from the KUMA 4.6 resource set. We recommend using the [OOTB] Kaspersky Industrial CyberSecurity for Networks 4.x syslog normalizer.
[OOTB] VipNet Coordinator Syslog Direction	Dictionary	The [OOTB] VipNet Coordinator Syslog Direction dictionary is obsolete and was removed from the KUMA 4.6 resource set.

Generating events for testing a normalizer

If necessary, you can generate your own example events to test your normalizer. Such testing makes it easier to write regular expressions and lets you see which values end up in the KUMA event fields.

Keep in mind the following special considerations:

- This tests simulates event processing. Example events in the **Example event** field are intended for displaying examples in the **Field mapping** section. Examples of the parent normalizer are used to generate examples of child normalizers, taking into account the **Field to pass into normalizer** setting.
- Mutations cannot be applied.

To test the normalizer, you need to add an example event to the **Event examples** field in the selected normalizer and start generating events by using the relevant command. As a result of running the command, KUMA takes the example event from the **Example event** field and sends events to the normalizer with the specified interval. If necessary, you can specify multiple examples to get events for multiple examples.

To test the normalizer:

1. Select the collector that you want to use for testing:
   - If the collector is installed on the server and running, stop the collector service:

     ```
     sudo systemctl stop kuma-collector-<collector service ID copied from the KUMA web interface>.service
     ```

   - If the collector is not running, or is in the process of being created or edited, proceed to the next step.

2. In the collector creation wizard, if necessary, fill in or edit the required fields at the **Connect event sources** step and at the **Transport** step, then proceed to the **Parsing** step:

   a. Link a normalizer by selecting it from the drop-down list, or create a normalizer.

   b. In the **Event examples** field, add example events. For example, for a json normalizer, you can add the following value: `{"name": "test_events", "address": "10.12.12.31"}`. You can specify multiple examples if you want to receive events for multiple examples in the same normalizer. Events are generated for each example.

3. In the Collector Installation Wizard, go to the **Routing** step and specify the storage where you want to save test events.

4. Review the collector settings and click **Save**.

5. Go to the **Active services** section in KUMA and click **Add** to add a collector. This opens the **Choose a service** window; in that window, select the collector and click **Create service**. The collector is displayed in the **Active services** list.

6. Check the status of the collector to which events are being sent. The collector status should be red.

7. Run the event generation command with the necessary parameters:

   - If the collector is not installed on the server, but only added in the **Active services** section:

     ```
     sudo /opt/kaspersky/kuma/kuma collector --core <FQDN of the KUMA Core server>:<port used by the KUMA Core for internal communication (port 7210 is used by default)> --generator.interval <interval in seconds for generating and sending events> --id <collector service ID copied from the KUMA web interface> --api.port <number of a free, unused API port>
     ```

     If the value of the event generation and sending interval is not specified or it is set to zero, events are not generated.

   - If the collector is installed on the server:

     ```
     sudo /opt/kaspersky/kuma/kuma collector --generator.interval <value of the event generation and sending interval in seconds> --id <collector service ID copied from the KUMA web interface> --api.port <number of a free, unused API port>
     ```

     If the value of the event generation and sending interval is not specified or it is set to zero, events are not generated.

   As a result, KUMA generates events and sends them to the normalizer, observing the specified interval.

   You can verify that events have been created and satisfy your expectations in the **Events** section. For additional information about the check, see the /etc/systemd/system/multi-user.target.wants/kuma-collector-<collector service ID copied from the KUMA web interface>.service file.

If the result does not meet expectations, modify the example event:

- If the collector is not installed on the server and has only been in the **Active services** section, edit the **Event examples** field in the normalizer of the collector and save the collector settings.

- If the collector is installed on the server and stopped as a service, edit the **Event examples** field in the normalizer of the collector, save the collector settings, go to the **Active services** section, select the collector, and refresh the collector settings by clicking **Refresh**.

If the result meets expectations:

1. Disable event generation, for example, by pressing **Ctrl+C** on the command line.

2. Start the collector service; if the service is already installed on the server, but has been stopped:

   ```
   sudo systemctl start kuma-collector-<collector service ID copied from the KUMA web interface>.service
   ```

3. If the collector has only been added in the **Active services** section, but has not been installed on the server yet, install the collector on the server using the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma collector --core <FQDN of the KUMA Core server>:<port used by KUMA Core server for internal communication (port 7210 by default)> --id <collector service ID copied from the KUMA web interface> --api.port <port used for communication with the installed component> --install
   ```

Glossary

Aggregation

Combining several messages of the same type from the event source into a single event.

Cluster

A group of servers on which the KUMA solution has been installed and that have been clustered together for centralized management using the solution's web interface.

Collector

KUMA component that receives messages from event sources, processes them, and transmits them to a storage, correlator, and/or third-party services to identify suspected information security incidents (alerts).

Connector

A KUMA component that ensures transport for receiving data from external systems.

Correlation rule

KUMA resource used to recognize defined sequences of processed events and perform specific actions after recognition.

Dashboard

Component of the KUMA system that performs data visualization.

Enrichment

The conversion of the textual representation of an event using dictionaries, constants, calls to the DNS service, and other tools.

Event

An instance of activity of network devices, application software, information security tools, operating systems, and other devices that can be detected and recorded. For example, events include: successful user logon events, log clear events, anti-virus software disable event.

Filter

The set of conditions the solution uses to select events for further processing.

KUMA web interface

A KUMA service that provides a user interface to configure and track KUMA operations.

Network port

A TCP and UDP protocol setting that defines the destination of IP-format data packets that are transmitted to a host over a network and allows various applications running on the same host to receive the data independently of each other. Each application processes the data sent to a specific port (sometimes it is said that the application listens to this port number).

It's standard practice to assign standard port numbers to certain common network protocols (for example, web servers usually receive data over HTTP on TCP port 80), although in general an application can use any protocol on any port. Possible values: from 1 to 65,535.

Normalization

A process that formats data received from an event in accordance with the fields of the KUMA event data model. During normalization, the data may be modified in accordance with certain rules (for example, changing upper case characters to lower case, replacing certain sequences of characters with others, etc.).

Normalizer

System component responsible for processing raw events from event sources. One normalizer processes events from one device or software of one specific version.

Parsing

The process of organizing data and converting incoming events into KUMA format.

Raw event

An event that has not passed the normalization stage in KUMA.

Report

KUMA resource that is used to generate a dataset based on user-defined filter criteria.

Role

A set of access privileges established to grant the KUMA web interface user the authority to perform tasks.

SELinux (Security-Enhanced Linux)

A system for controlling process access to operating system resources based on the use of security policies.

SIEM

Security Information and Event Management system. A solution for managing information and events in a company's security system.

STARTTLS

Text exchange protocol enhancement that lets you create an encrypted connection (TLS or SSL) directly over an ordinary TCP connection instead of opening a separate port for the encrypted connection.

UserPrincipalName

UserPrincipalName (UPN)—user name in email address format, such as `username@domain.com`.

The UPN must match the actual email address of the user. In this example, `username` is the user name in the Active Directory domain (user logon name), and `domain.com` is the UPN suffix. They are separated by the `@` character. The DNS name of the Active Directory domain is used as the default UPN suffix in Active Directory.

Information about third-party code

Information about third-party code is in the LEGAL_NOTICES file located in the /opt/kaspersky/kuma/LEGAL_NOTICES folder.

Trademark notices

Registered trademarks and service marks are the property of their respective owners.

AMD is a trademark or registered trademark of Advanced Micro Devices, Inc.

Apache, Apache Cassandra, Apache Tomcat are either registered trademarks or trademarks of the Apache Software Foundation in the United States and/or other countries.

Ubuntu, LTS are registered trademarks of Canonical Ltd.

Cisco, IOS, Snort are registered trademarks or trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

Citrix, Citrix NetScaler are either a registered trademark or a trademark of Cloud Software Group, Inc., and/or its subsidiaries in the United States and/or other countries.

Dameware, SolarWinds are trademarks that are the exclusive property of SolarWinds Worldwide, LLC or its affiliates, are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries.

Dell Technologies, Dell, and other trademarks are trademarks of Dell Inc or its subsidiaries.

The Grafana word mark and the 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. We are not affiliated with, endorsed, or sponsored by Coding Instinct, or the Grafana community.

Elasticsearch is a trademark of Elasticsearch BV, registered in the U.S. and in other countries.

ESET is 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, FortiOS, FortiSOAR are either registered trademarks or trademarks of Fortinet, Inc. in the United States and/or other countries.

The FreeBSD mark is a registered trademark of The FreeBSD Foundation.

Google, Chrome, Google Chrome are trademarks of Google LLC.

HUAWEI, Huawei Eudemon are trademarks of Huawei Technologies Co., Ltd.

IBM, Guardium, InfoSphere are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide.

Intel, Core are trademarks of Intel Corporation or its subsidiaries.

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.

Mac is a trademark of Apple Inc.

Microsoft, Active Directory, AppLocker, Azure, Excel, Halo, Hyper-V, Lync, Office 365, Outlook, PowerShell, SharePoint, Skype, SQL Server, Windows, Windows PowerShell, and Windows Server are trademarks of the Microsoft group of companies.

Mozilla and Firefox are trademarks of the Mozilla Foundation in the United States and other countries.

NetApp is a trademark or 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.

OpenSSL is a trademark owned by the OpenSSL Software Foundation.

OpenVPN is a registered trademark of OpenVPN, Inc.

Oracle is a registered trademark of Oracle and/or its affiliates.

Python is a trademark or registered trademark of the Python Software Foundation.

Red Hat, Red Hat Enterprise Linux, CentOS, Fedora are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

Ansible is a registered trademark of Red Hat, Inc. in the United States and other countries.

Sendmail, and other product designations or names are trademarks or registered trademarks of Sendmail, Inc.

Debian is a registered trademark of Software in the Public Interest, Inc.

The CommuniGate Pro name is a trademark or registered trademark of Stalker Software, Inc.

Symantec is a trademark or registered trademark of Symantec Corporation or its affiliates in the U.S. and other countries.

OpenAPI is a trademark of The Linux Foundation.

Kubernetes is a registered trademark of The Linux Foundation in the United States and other countries.

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.

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.

ViPNet is a registered trademark of Infotecs.

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.

Known limitations and issues

KUMA has some limitations that are not critical for the operation of the solution:

- After upgrading KUMA 3.4 to version 4.0.1 in the **Settings** → **Other** → **Services monitoring** section, the **Detached parts threshold, %** setting is reset to 0.

- After upgrading KUMA 3.4 to version 4.0, some widgets on the dashboard are displayed with an error. To fix the error, please manually refresh the data on the dashboard or wait for it to be refreshed automatically.

- The widget form can be opened below the **Dashboards** settings form when you follow links from the **Used in widgets** column if you have opened the settings of any widget before opening the **Dashboards** settings form.