kaspersky

Kaspersky Unified Monitoring and Analysis Platform

Contents

Kaspersky Unified Monitoring and Analysis Platform Help

New features

- What's new in KUMA

Hardware and software requirements

- Hardware and software requirements

Getting started

- Program architecture
- 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 "program") is an integrated software solution that includes the following set of functions:

- Receiving, processing, and storing information security events.
- Analysis and correlation of incoming data.
- Search within the obtained events.
- Creation of notifications upon detecting symptoms of information security threats.

The program is built on a microservice architecture. This means that you can create and configure the relevant microservices (hereinafter also "services"), thereby making it possible to use KUMA both as a log management system and as a full-fledged SIEM system. In addition, flexible data streams routing allows you to use third-party services for additional event processing.

What's new

The new version of Kaspersky Unified Monitoring and Analysis Platform introduces the following features and improvements:

- Added support for the following operating systems:
  - Oracle Linux 9.2
  - Astra Linux 1.7.4
- Filter conditions and correlation rules can be written as code.

  Redesigned interface for writing conditions in filters and correlation rule selectors. In addition to the builder mode, a new mode allows writing conditions as code. Filter and selector conditions are automatically translated between builder mode and code mode.

  - The builder allows writing conditions using the keyboard. As you start typing filter conditions, KUMA suggests matching options, for example, from event fields ("e:"), dictionaries ("d:"), active lists ("a:"), after which you can select the option you want. You can immediately narrow down the range of options by typing the appropriate prefix ("e:SourceAddress"). Condition types are highlighted in different colors.
  - Code mode lets you quickly edit conditions, select and copy blocks of code to move filters or selector conditions between filters and correlation rules. Linked resources are automatically named. Fields with the names of linked resources are not available for editing.
- Extended event schema.

  In KUMA 3.0.2, CEF remains the basis of the KUMA event schema, but now you can create custom fields. This functionality allows you to implement an arbitrary taxonomy. Field names that are familiar to users help write search queries quicker. Custom fields are typed and must begin with a prefix that determines the type of the field: S for string, N for numeric, F for floating-point. Array types are SA for an array of strings, NA for an array of numbers, FA for an array of floating-point numbers. Array fields can be used only in JSON and KV normalizers.
- Automatic identification of the event source.

  The following event normalization options are now available:

1 collector — 1 normalizer

We recommend using this method if you have many events of the same type or many IP addresses from which events of the same type may originate. You can configure one collector with only one normalizer, which is optimal in terms of performance.

1 collector — multiple normalizers linked to IP

This method is available for collectors with a connector of UDP, TCP, or HTTP type. If a UDP, TCP, or HTTP connector is specified in the collector at the 'Transport' step, then at the 'Event parsing' step, you can specify multiple IP addresses on the 'Parsing settings' tab and choose the normalizer that you want to use for events coming from the specified addresses. The following types of normalizers are available: json, cef, regexp, syslog, csv, kv, xml. For normalizers of the Syslog and regexp types, you can specify extra normalization conditions depending on the value of the DeviceProcessName field.

- Linking a correlator from the rule list section.

  Now you can publish correlation rules on correlators using the correlation rule menu. Now you can select one or more rules and link them to one or more correlators. The list of correlators is represented by a tree structure of correlator resources. An analyst managing correlation rules of the Shared tenant can publish rules on correlators of all tenants available to that analyst.

- Using context tables.

  Context tables are a new entity in the correlator. You can manage context tables in a similar way as active lists. Functionality of context tables:

  - The list of key fields is defined by the user.
  - Data in context tables is typed (integer, float, string, boolean, timestamp, IP).
  - Arrays are supported for all data types listed above.
  - Array fields of context tables can be used in correlation rules: you can count unique values, calculate the length of the array, or reference a specific element of the array.

- The kata/edr connector for telemetry from KEDR hosts.

  KUMA supports obtaining KEDR events using the kata/edr connector. The kata/edr connector differs from kafka in that it allows integration with KATA 5.1, and configuring it takes less time and effort. You can use collector and filter settings to make the request more specific—you can specify how many events you want to receive in one request, which events and how frequently you want to receive.

- Archiving and deleting assets.

  Assets imported from KSC or KICS whose agents no longer connect to KSC or KICS, are marked as archived in KUMA and can be deleted after a user-defined period.

- The FQDN field in assets can contain an array of values.

- The CEF format is added for sending events to third-party systems.

- 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.

- Predefined layouts for the main B2B products by Kaspersky (KATA & EDR, KSC, KSMG, KWTS) let you visualize data as soon as the corresponding event sources are connected.

- Hierarchical structure is no longer supported. When upgrading from version 2.1.3 to 3.0.2, all KUMA hosts become stand-alone.

- Now you can send test events.

- Integration with Telegram, UserGate, KWTS, KSMG, RedCheck is now supported.

- Logs: When upgrading from 2.1.3 to 3.0.2, the old Core logs are preserved and the new Core logs are created beside them. After the upgrade, the list of logs looks as follows:

  - /opt/kaspersky/kuma/core/log/core — version 2.1.3 logs.

  - /opt/kaspersky/kuma/core/log/stdout.log — version 3.0.2 logs; the standard output stream of the service is redirected here.

  - /opt/kaspersky/kuma/core/log/stderr.log — version 3.0.2 logs; the standard error stream of the service is redirected here.

- Role-based access model was changed.

  - Administrator → Tenant administrator: can now delete predefined report templates and predefined dashboard layouts.

  - Analyst → Tier 2 analyst: can now view and generate all reports, their own and those of other users, provided that all tenants specified in the report template are available to this role. Can edit predefined templates, edit the predefined report generation settings, can delete predefined report templates and predefined dashboard layouts.

  - First line analyst → Tier 1 analyst: can now view and generate all reports, their own and those of other users, provided that all tenants specified in the report template are available to this role.

  - Operator → Junior analyst

  New roles: Access to shared resources, Interaction with NCIRCC, Interaction with CII.

  When kuma-core.service is restarted, the predefined report templates and dashboard layout templates are restored to their original condition if they were previously deleted.

  Specifying the user's email address in the template is no longer grounds for providing access to a report generated from that template. Such a report is available to the user for viewing if all tenants specified in the template are available for the user's role.

- Normalizers for the event sources are added.

- The Appendices section provides a list of deprecated KUMA resources and a possible way to replace them.

- Bugs of previous versions were fixed.

Distribution kit

The distribution kit includes the following files:

- kuma-ansible-installer-<build number>.tar.gz is used to install KUMA components without the capability of deployment in a high availability configuration.

- kuma-ansible-installer-ha-<build number>.tar.gz is used to install KUMA components with the capability of deployment in a high availability configuration.
- files containing information about the version (release notes) in Russian and English.

Hardware and software requirements

Recommended hardware requirements

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

The following table lists the hardware and software requirements of KUMA components. 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 instruction set support.

	Up to 3,000 EPS	Up to 10,000 EPS	Up to 20,000 EPS	Up to 50,000 EPS
Configuration	Installation on a single server 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 100 Mbps.	Installation on a single server One device. Device characteristics: At least 24 threads or 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.	1 server for the Core + 1 server for the Collector + 1 server for the Correlator + 3 dedicated servers with the Keeper role + 2 servers for the Storage	1 server for the Core + 2 servers for the Collector + 1 server for the Correlator + 3 dedicated servers with the Keeper role + 4 servers for the Storage
Requirements for the Core component	-	-	One device. Device characteristics: At least 10 threads or vCPUs. At least 24 GB of RAM. At least 500 GB in the /opt directory.	One device. Device characteristics: At least 10 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 100 Mbps.	Data storage type: SSD. Data transfer rate: at least 100 Mbps.
Requirements for the Collector component	-	-	One device. Device characteristics: At least 8 threads or vCPUs. At least 16 GB of RAM. At least 500 GB in the /opt directory. Data storage type: HDD allowed. Data transfer rate: at least 100 Mbps.	Two devices. Characteristics of each device: At least 8 threads or vCPUs. At least 16 GB of RAM. At least 500 GB in the /opt directory. Data storage type: HDD allowed. Data transfer rate: at least 100 Mbps.
Requirements for the Correlator component	-	-	One device. Device characteristics: At least 8 threads or vCPUs. At least 32 GB of RAM. At least 500 GB in the /opt directory. Data storage type: HDD allowed. Data transfer rate: at least 100 Mbps.	One device. Device characteristics: At least 8 threads or vCPUs. At least 32 GB of RAM. At least 500 GB in the /opt directory. Data storage type: HDD allowed. Data transfer rate: at least 100 Mbps.
Requirements for the Keeper component	-	-	Three devices. Characteristics of each device: At least 6 threads or vCPUs. At least 12 GB of RAM. At least 150 GB in the /opt directory. Data storage type: SSD. Data transfer rate: at least 100 Mbps.	Three devices. Characteristics of each device: At least 6 threads or vCPUs. At least 12 GB of RAM. At least 150 GB in the /opt directory. Data storage type: SSD. Data transfer rate: at least 100 Mbps.
Requirements for the Storage component	-	-	Two devices. Characteristics of each device: At least 24 threads or vCPUs. At least 64 GB of RAM.	Four devices. Characteristics of each device: At least 24 threads or vCPUs. At least 64 GB of RAM.

			At least 500 GB 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.	At least 500 GB 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	• Oracle Linux 8.6, 8.7, 9.2. • Astra Linux Special Edition RUSB.10015-01 (2021-1126SE17 update 1.7.1). • Astra Linux Special Edition RUSB.10015-01 (2022-1011SE17MD update 1.7.2.UU.1). • Astra Linux Special Edition RUSB.10015-01 (2022-1110SE17 update 1.7.3). Core version 5.15.0.33 or higher is required. • Astra Linux Special Edition RUSB.10015-01 (2023-0630SE17MD, update 1.7.4.UU.1).			

Depending on the number and complexity of database queries made by users, reports, and dashboards, a greater amount of resources may be required.

For every 50,000 (above 50,000) assets, you must add 2 extra threads or vCPUs and 4 GB of RAM to the resources of the Core component.

For every 100 (above 100) services managed by the Core component, you must add 2 additional threads or vCPUs to the resources of the Core component.

ClickHouse must be deployed on solid-state drives (SSD). SSDs help improve data access speed.

* if the usage profile of the system does not involve deep SQL aggregate queries to Storage, HDD-based disk arrays may be used.

Hard drives can be used to store data using the HDFS technology.

Exported events are written to the drive of the Core component to the /opt/kaspersky/kuma/core/tmp/ temporary folder. The exported data is stored for 10 days and then automatically deleted. If you plan to export a large amount of events, you must allocate additional space.

Working in virtual environments

Installation of KUMA is supported in the following virtual environments:

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

Resource recommendations for the Collector component

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 1,000 EPS and event enrichment disabled (event enrichment is disabled, event aggregation is disabled, 5,000 accounts, 5,000 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.

Kaspersky recommendations for storage servers

To connect a data storage system to storage servers, you must use high-speed protocols, such as Fibre Channel or iSCSI 10G. We do not recommend using application-level protocols such as NFS and SMB to connect data storage systems.

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

If you are using RAID arrays, it is recommended to use 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 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. In this case, it is necessary to prevent two virtual machines with ClickHouse from working on the same hypervisor.

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

Requirements for devices for installing agents

To have data sent to the KUMA collector, you must install agents on the network infrastructure devices. 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 Server® 2012 R2 • Microsoft Windows Server 2016 • Microsoft Windows Server 2019 • Microsoft Windows 10 20H2, 21H1	• Oracle® Linux 8.6, 8.7, 9.2. • Astra Linux Special Edition RUSB.10015-01 (2021-1126SE17 update 1.7.1). • Astra Linux Special Edition RUSB.10015-01 (2022-1011SE17MD update 1.7.2.UU.1). • Astra Linux Special Edition RUSB.10015-01 (2022-1110SE17 update 1.7.3). • Astra Linux Special Edition RUSB.10015-01 (2023-0630SE17MD, update 1.7.4.UU.1).

Requirements for client devices for managing the KUMA web interface

CPU: Intel® Core™ i3 8th generation

RAM: 8 GB

Supported browsers:

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

Device requirements for installing KUMA on Kubernetes

The minimum configuration of a Kubernetes cluster for deployment of a high availability KUMA configuration includes the following:

- 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.

	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	At least 1 TB in the /opt directory.

			At least 32 GB in the /var/lib directory.
Network bandwidth	10 Gbps	10 Gbps	10 Gbps

KUMA interface

The program is managed through the web interface.

The window of the program web interface contains the following items:

- Sections in the left part of the program web interface window
- Tabs in the upper part of the program web interface window for some sections of the program
- Workspace in the lower part of the program web interface window

The workspace displays the information that you choose to view in the sections and on the tabs of the program web interface window. It also contains management elements that you can use to configure how the information is displayed.

While working with the program web interface, you can use hot 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

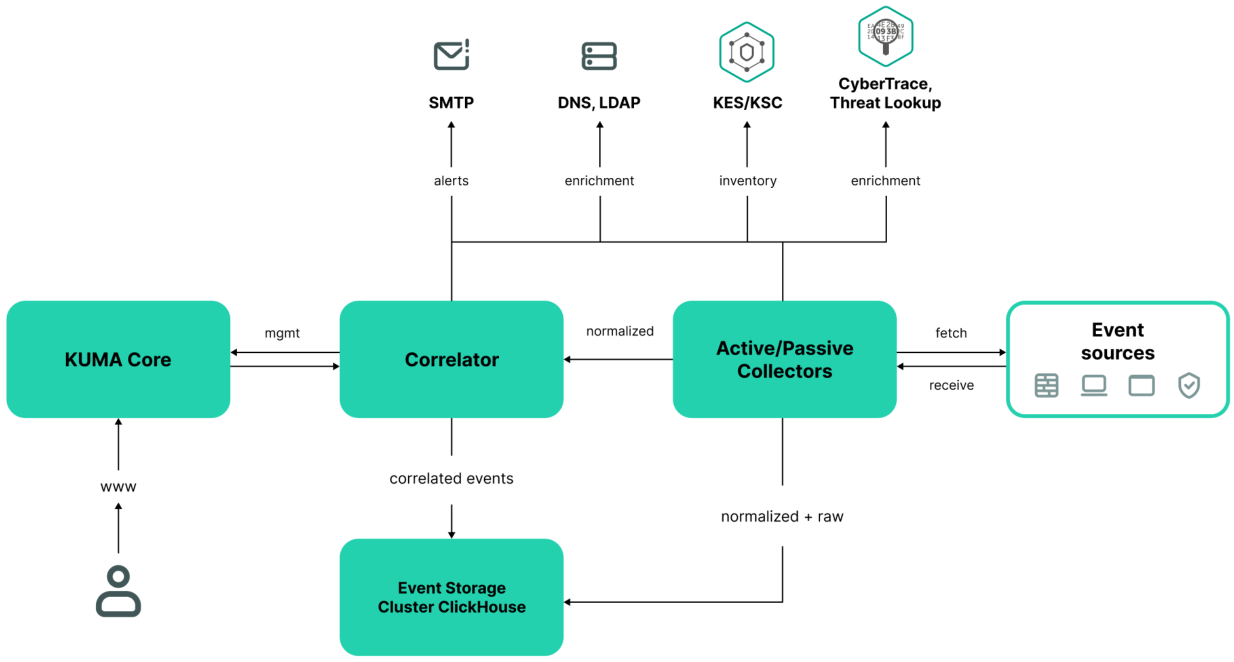
If the components of KUMA and Kaspersky Endpoint Security for Linux are installed on the same server, the report.db directory may grow very large and even take up the entire drive space. To avoid this problem, it is recommended to upgrade Kaspersky Endpoint Security for Linux to version 11.2 or later.

Program architecture

The standard program installation includes the following components:

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

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



KUMA architecture

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 program, as well as integrate the necessary software into the system;

- manage program services and user accounts in a centralized way;
- visualize statistical data on the program;
- investigate security threats based on the received events.

Collector

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

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

The collector operation algorithm includes the following steps:

1. **Receiving messages from event sources**

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

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

   The program has the following connector types available:

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

- snmp

2 Event parsing and normalization

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

The following normalizers are available in the program:

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

3 Filtering of normalized events

You can configure filters that allow you to filter out events that meet specified conditions. Events that do not meet the filtering conditions will be sent for processing.

4 Enrichment and conversion of normalized events

Enrichment rules let you to supplement event contents with information from internal and external sources. The program has 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 program has the following conversion methods:

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

5 **Aggregation of normalized events**

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

6 **Transmission of normalized events**

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

Correlator

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

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

- Alert detection.
- Notification about detected incidents.
- Active lists content management.
- Sending correlation events to configured destinations.

Event correlation is performed in real time. The operating principle of the correlator is based on an event signature analysis. This means that every event is processed according to the correlation rules set by the user. When the program detects a sequence of events that satisfies the conditions of the correlation rule, it creates a correlation event and sends it to the Storage. The correlation event can also be sent to the correlator for repeated analysis, which allows you to customize the correlation rules so that they are triggered by the results of a previous analysis. 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 sharing the load between services. In this case, the collectors will send normalized events to all available correlators.

The correlator operation algorithm has the following steps:

1. **Obtaining an event**

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

2. **Applying correlation rules**

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

3. **Responding to an alert**

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

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

4. **Sending a correlation event**

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

Storage

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

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

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

Basic entities

This section describes the main entities that KUMA works with.

About tenants

KUMA has a multitenancy mode in which one instance of the KUMA application 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 main interface while tenants operate independently of each other and have access only to their own resources, services, and settings. Events of tenants are stored separately.

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

In KUMA, two tenants are created by default:

- The main tenant contains resources and services related to the main tenant. These resources are available only to the general administrator.
- The shared tenant is where the general administrator can place resources, asset categories, and monitoring policies that users of all tenants will be able to utilize. Access to a shared tenant can be **restricted** to individual users.

  > If in user settings, the **Hide shared resources** check box is selected, the user cannot gain access to the Shared folder belonging to the shared tenant in the KUMA web interface in the **Resources** → **<resource type>**. 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 user can see the names of these resources in the service settings, but 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.

About events

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

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

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

The first steps are carried out in a collector.

1. Raw event. The original message received by KUMA from an event source using a Connector is called a *raw event*. This is an unprocessed message and it cannot be used yet by KUMA. To fit into the KUMA pipeline, raw events must be normalized into the KUMA data model. That's what the next stage is for.
2. Normalized event. A normalizer transforms 'raw' event data in accordance with the KUMA data model. After this conversion, the original message becomes a *normalized event* and can be used by KUMA for analysis. From here on, only normalized events are used in KUMA. Raw events are no longer used, but they can be kept as a part of normalized events inside the `Raw` field.

   The program has the following normalizers:

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

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

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

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

Event types:

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

About alerts

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

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

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

You can create incidents based on alerts.

Alert management in KUMA is described in this section.

About incidents

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

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

Incidents can be exported to NCIRCC.

About assets

Assets are network devices registered in KUMA. Assets generate network traffic when they send and receive data. The KUMA program can be configured to track this activity and create baseline 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 contain this IP address in any of its parameters.

Assets can be divided into logical groups. 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 high severity to a certain category of assets, the base events involving these assets will trigger the creation of correlation events with higher severity. This in turn cascades into higher severity alerts and, therefore, a more rapid response to such an alert.

It is recommended to register network assets in KUMA because their use makes it possible to formulate clear and versatile correlation rules for much more efficient analysis of events.

Asset management in KUMA is described in this section.

About 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.

About services

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

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

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

About agents

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

Types of agents:

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

About Priority

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

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

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

Possible severity values:

- Low
- Medium
- High
- Critical

Program licensing

This section covers the main aspects of program 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 program.

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

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

- Go to the folder with the unpacked installer and read the ./roles/kuma/files/LICENSE text file.
- Go to the folder with the unpacked 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 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 'kuma' executable file and run the following command to display the text of the End User License Agreement:

  ```
  ./kuma license --show
  ```

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

About the license

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

A license entitles you to the following kinds of services:

- Use of the program 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 program was activated.

A license is provided when the program is purchased. When the license expires, the program continues to work but with limited functionality (for example, new resources cannot be created). 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 key or order number
- Information about the user who is granted the license
- Information about the program 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 program in accordance with the terms of the End User License Agreement. License keys are generated by Kaspersky specialists.

You can add a license key to the program by applying a *key file*. The license key is displayed in the program interface as a unique alphanumeric sequence after you add it to the program.

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 program.

A license key may be active or reserve.

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

A *reserve license key* is the license key that entitles the user to use the program 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 program.

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 program 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.

Data provision in Kaspersky Unified Monitoring and Analysis Platform

Data provided to third parties

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

Locally processed data

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

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

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

- Information about devices on the corporate network.

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

  - Import assets:

    - On demand from MaxPatrol.

    - On a schedule from Kaspersky Security Center and KICS for Networks.

  - Create assets manually through the web interface or via the API.

  KUMA stores the following device information:

  - Technical characteristics of the device.

  - 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 monitored object, attacked network service, and port/protocol.

- 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, connection credentials, and certificate in the KUMA console.

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

- Information contained in events from configured sources.

  In the collector, the event source is configured, KUMA events are generated and sent to other KUMA services. Sometimes events can arrive first at the agent service, which relays events from the source to the collector.

- 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, R-Vision Security Orchestration, Automation and Response).

  It can include certificates, tokens, URLs or credentials for establishing a connection with the other application, or other data necessary for the basic functionality of KUMA, for example, email. The user enters this data in the KUMA console

- Information about sources from which event receipt is configured.

  It can include the source name, host name, IP address, the monitoring policy assigned to the source. The monitoring policy specifies the email address of the person responsible, to whom a notification will be sent if the policy is violated.

- User accounts: name, username, email address. The user can view their profile data in the KUMA console.

- User profile settings:

    - User role in KUMA. Users can see their assigned roles.
    - Localization language, notification settings, display of non-printable characters.

      The user enters this data in the KUMA interface.
    - List of asset categories in the **Assets** section, default dashboard, TV mode flag for the dashboard, SQL query for default events, default preset.

      The user specifies these settings in the corresponding sections of the KUMA console.
- Audit events

  KUMA automatically records audit events.
- KUMA log

  The user can enable extended logging in the KUMA console. Log entries are stored on the user's device, no data is transmitted automatically.
- Information about the user accepting the terms and conditions of legal agreements with Kaspersky.
- Any information that the user enters in the KUMA interface.

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

- The user enters information in the KUMA console.
- KUMA services (agent or collector) receive data if the user has configured a connection to event sources.
- Through the KUMA REST API.
- Device information can be obtained using the utility from MaxPatrol.

The listed information is stored in the KUMA database (MongoDB, ClickHouse). 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 to the program web interface

You can add an application license key in the KUMA web interface.

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

To add a license key to the KUMA web interface:

1. Open the KUMA web interface and select **Settings** → **License**.

   The window with KUMA license conditions opens.

2. Select the key you want to add:

   - If you need to add an active key, click the **Add active license key** button.

     This button is not displayed if a license key has already been added to the program. If you want to add an active license key instead of the key that has already been added, the current license key must be deleted.

   - If you want to add a reserve key, click the **Add reserve license key** button.

     This button is inactive until an active key is added. If you want to add a reserve license key instead of the key that has already been added, the current reserve license key must be deleted.

   The license key file selection window appears on the screen.

3. Select a license file by specifying the path to the folder and the name of the license key file with the KEY extension.

The license key from the selected file will be loaded into the program. Information about the license key is displayed under **Settings** → **License**.

Viewing information about an added license key in the program web interface

In the KUMA web interface, you can view information about the added license key. Information about the license key is displayed under **Settings** → **License**.

Only users with the Administrator role can view license information.

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

- **Expires on**—date when the license key expires.
- **Days remaining**—number of days before the license is expired.
- **EPS available**—number of events processed per second supported by the license.
- **EPS current**—current average number of events per second processed by KUMA.
- **License key**—unique alphanumeric sequence.
- **Company**—name of the company that purchased the license.
- **Client name**—name of client who purchased the license.
- **Modules**—modules available for the license.

Removing a license key in the program web interface

In KUMA, you can remove an added license key from the program (for example, if you need to replace the current license key with a different key). After the license key is removed, the program stops to receive and process events. This functionality will be re-activated the next time you add a license key.

Only users with the administrator role can delete license keys.

To delete an added license key:

1. Open the KUMA web interface and select **Settings** → **License**.

   The window with KUMA license conditions opens.

2. Click the 🗑 icon on the license that you want to delete.

   A confirmation window opens.

3. Confirm deletion of the license key.

The license key will be removed from the program.

Administrator's guide

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

Installing and removing KUMA

To complete the installation, you need a distribution kit:

- **kuma-ansible-installer-<build number>.tar.gz** contains all necessary files for installing KUMA without the support for high availability configurations.
- **kuma-ansible-installer-ha-<build number>.tar.gz** contains all necessary files for installing KUMA in a high availability configuration.

To complete the installation, you need the install.sh installer file and an inventory file that describes the 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

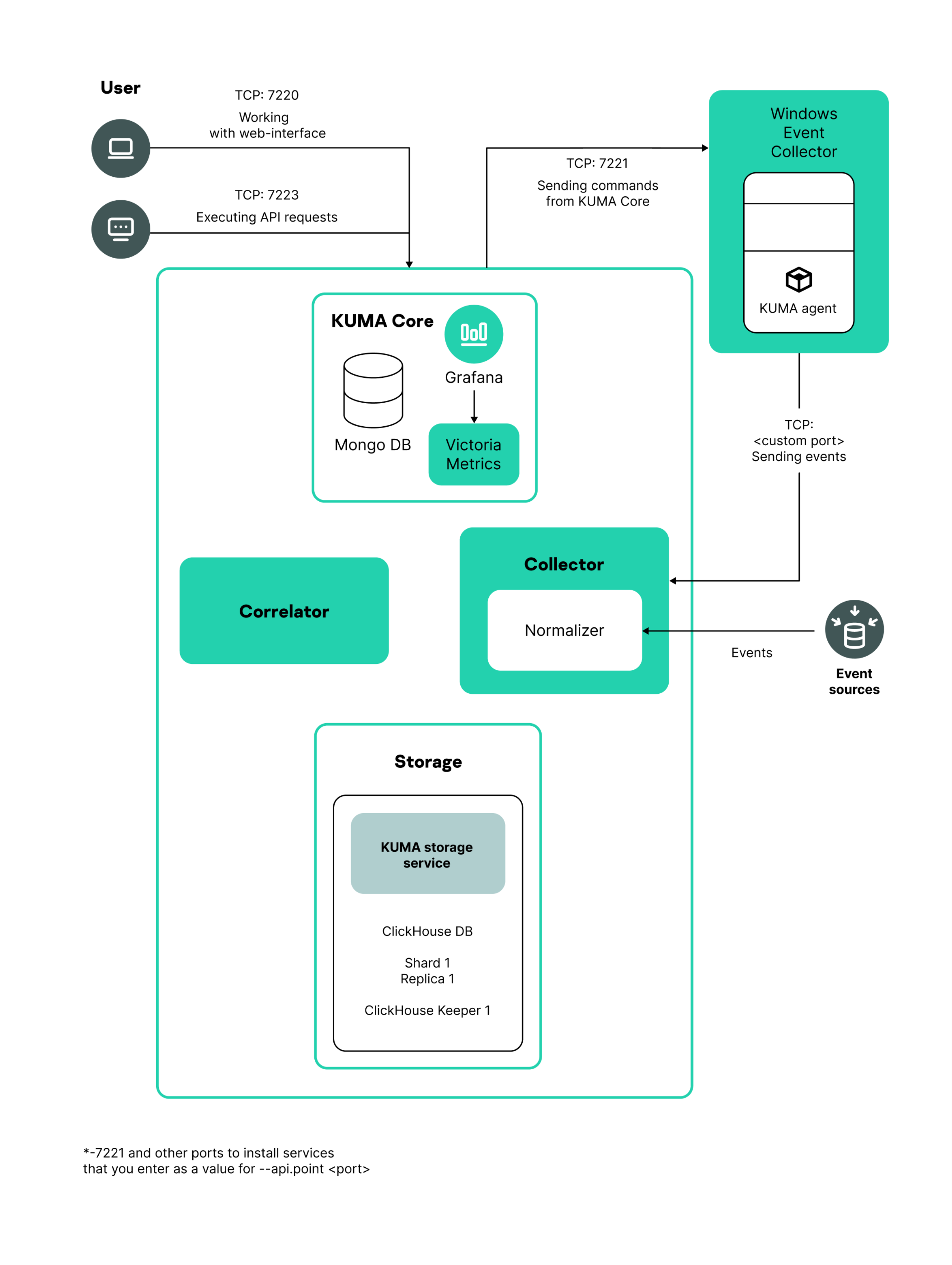
KUMA places its files in the /opt directory, so we recommend making /opt a separate partition and allocating 16 GB for the operating system and the remainder of the disk space for the /opt partition.

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



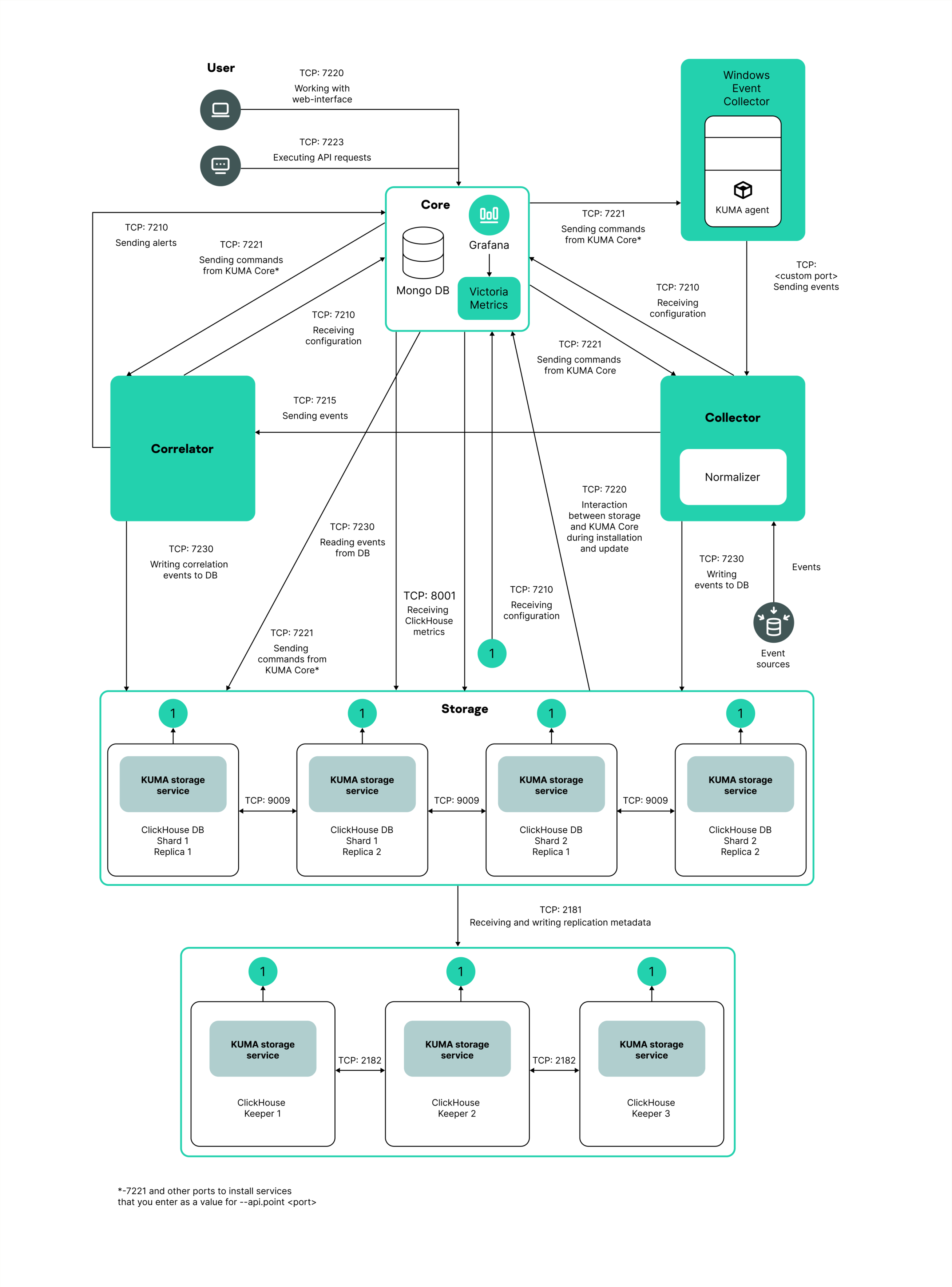
*-7221 and other ports to install services
that you enter as a value for --api.point <port>

Installation on a single server

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 multiple dashboard layouts and handle a high volume of search queries, a single server might not be sufficient. In that case, we recommend choosing the distributed installation instead.

- Distributed installation

Distributed Installation diagram



Distributed Installation diagram

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

- Distributed installation in a high availability configuration

  You can install the KUMA Core on a Kubernetes cluster for high availability. Use the k0s.inventory.yml inventory file for the description.

Program installation requirements

General application installation requirements

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

- Servers on which you want to install the components satisfy the hardware and software requirements.
- Ports used by the installed instance of KUMA are available.
- KUMA components are addressed using the fully qualified domain name (FQDN) of the host. Before you install the application, make sure that the correct host FQDN is returned in the `Static hostname` field. For this purpose, execute the following command:

  ```
  hostnamectl status
  ```

- The server where the installer is run does not have the name `localhost` or `localhost.<domain>`.
- Time synchronization over Network Time Protocol (NTP) is configured on all servers with KUMA services.

Installation requirements for Oracle Linux and Astra Linux operating systems

	Oracle Linux	Astra Linux
Python version	3.6 or later	3.6 or later
SELinux module	Disabled	Disabled
Package manager	pip3	pip3
Basic packages	• netaddr • firewalld • compat-openssl11 — installing this package is necessary on a host with Oracle Linux 9 where the KUMA Core must be deployed outside a cluster. See detailed information about upgrading from Oracle Linux 8.x to Oracle Linux 9.x The packages can be installed using the following commands: `pip3 install netaddr`	• python3-apt • curl • libcurl4 The packages can be installed using the following command: `apt install python3-apt curl libcurl4`

	`yum install firewalld`
	`yum install compat-openssl11`	

Dependent packages	–	- netaddr - python3-cffi-backend The packages can be installed using the following command: `apt install python3-netaddr python3-cffi-backend` If you are planning to query Oracle DB databases from KUMA, you must install the libaio1 Astra Linux package.
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	- libgtk2.0.0 - libnss3 - libatk-adaptor - libatk-1.0.so.0 - libdrm-common - libgbm1 - libxkbcommon0 - libasound2
User permissions level required to install the application	–	To assign the required permissions to the user account used for installing the application, run the following command: `sudo pdpl-user -i 63 <user name under which the application is being installed>`

Upgrading from Oracle Linux 8.x to Oracle Linux 9.x

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-grafana.service`
   - `sudo systemctl disable kuma-mongodb.service`
   - `sudo systemctl disable kuma-victoria-metrics.service`
   - `sudo systemctl disable kuma-vmalert.service`
   - `sudo systemctl disable kuma-core.service`
2. Update the OS on every host.
3. After the update, run the following command to install the compat-openssl11 package on the host where you want to deploy the KUMA Core outside a cluster:

   ```
   yum install compat-openssl11
   ```
4. Run the following commands to enable the services on the hosts where the services are installed:
   - `sudo systemctl enable kuma-core.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-grafana.service`
   - `sudo systemctl enable kuma-mongodb.service`
   - `sudo systemctl enable kuma-victoria-metrics.service`
   - `sudo systemctl enable kuma-vmalert.service`
5. Restart the hosts.

As a result, the upgrade is completed.

Ports used by KUMA during installation

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

Before installing the Core on the device, make sure that the following ports are free:

- 9090: used by Victoria Metrics.
- 8880: used by VMalert.
- 27017: used by MongoDB.

The table below shows the default network ports values. The installer automatically opens the ports during KUMA installation

Network ports used for the interaction of KUMA components

Protocol	Port	Direction	Destination of the connection
HTTPS	7222	From the KUMA client to the server with the KUMA Core component.	Reverse proxy in the CyberTrace system.
HTTPS	8123	Local requests from the storage service to the local node of the ClickHouse cluster.	Writing and receiving normalized events in the ClickHouse cluster.
HTTPS	9009	Between ClickHouse cluster replicas.	Internal communication between ClickHouse cluster replicas for transferring data of the cluster.
TCP	2181	From ClickHouse cluster nodes to the ClickHouse keeper replication coordination service.	Receiving and writing of 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.	Receipt of the configuration by KUMA from the KUMA Core server.
TCP	7220	• From the KUMA client to the server with the KUMA Core component. • From storage hosts to the server with the KUMA Core component 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 the port after the installation or upgrade.
TCP	7221 and other ports used for service installation as the --	From KUMA Core to KUMA services.	Administration of services from the KUMA web interface.

	api.port <port> parameter value		
TCP	7223	To the KUMA Core server.	Default port used for API requests.
TCP	8001	From Victoria Metrics to the ClickHouse server.	Receiving ClickHouse server operation metrics.
TCP	9000	From the local client.sh client to the local cluster node.	Writing and receiving data in the ClickHouse cluster.

Ports used by the OOTB predefined resources

The installer automatically opens the ports during KUMA installation.

Ports used by the OOTB predefined 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 "KUMA Core traffic in a high availability configuration" table shows the initiator of the connection (the source) 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
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

Synchronizing time on servers

To configure time synchronization on servers:

1. Install chrony:

   ```
   sudo apt install chrony
   ```

2. Configure the system time to synchronize with the NTP server:

   a. Make sure the virtual machine has Internet access.

      If access is available, go to step b.

      If internet access is not available, edit the `/etc/chrony.conf` file to replace `2.pool.ntp.org` with the name or IP address of your organization's internal NTP server.

   b. Start the system time synchronization service by executing the following command:

      ```
      sudo systemctl enable --now 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 the line "System clock synchronized: yes".

Synchronization is configured.

About the inventory file

KUMA components can be installed, updated, and removed from the directory with the unpacked kuma-ansible-installer using the Ansible tool and the inventory file you created. You can specify values for KUMA configuration settings in the inventory file; the installer uses these values when deploying, updating, and removing the program. The inventory file uses the YAML format.

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

- single.inventory.yml.template—Used to install KUMA on a single server. It contains the minimum set of settings optimized for installation on a single device without the use of a Kubernetes cluster.
- distributed.inventory.yml.template—Used for the initial distributed installation of KUMA without using a Kubernetes cluster, for expanding the all-in-one installation to a distributed installation, and for updating KUMA.
- expand.inventory.yml.template—Used in some reconfiguration scenarios: for adding collector and correlator servers, for expanding an existing storage cluster, and for 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—Used to install or migrate KUMA to a Kubernetes cluster.

We recommend backing up the inventory file that you used to install the program. 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 or an ipv4 or ipv6 IP address.

Example:

```
hosts:
 hostname.example.com:
 ip: 0.0.0.0
 or
 ip: ::%eth0
```

all block

In this block the variables that are applied to all hosts indicated in the inventory are specified, including the implicit localhost where the installation is started. Variables can be redefined at the level of host groups or even for individual hosts.

Example of redefining 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 following table lists possible variables in the 'vars' section and their descriptions.

List of possible variables in the vars section

Variable	Description	Possible values
`ansible_connection`	Method used to connect to target machines.	- `ssh`—connection to remote hosts via SSH. - `local`—no connection to remote hosts is established.
`ansible_user`	User name used to connect to target machines and install components.	If the root user is blocked on the target machines, use a user name that has the right to establish SSH connections and elevate privileges using su or sudo.

`ansible_become`	Indicates the need to increase the privileges of the user account that is used to install KUMA components.	`true` if the ansible_user value is not `root`.
`ansible_become_method`	A method for increasing the privileges of the user account that is used to install KUMA components.	`su` or `sudo` if the ansible_user value is not `root`.
`ansible_ssh_private_key_file`	Path to the private key in the format /<path>/.ssh/id_rsa. This variable must be defined if you need to specify a key file that is different from the default key file: ~/.ssh/id_rsa.	
`deploy_to_k8s`	Indicates that KUMA components are deployed in a Kubernetes cluster.	• `false` is the default value for the single.inventory.yml and distributed.inventory.yml templates. • `true` – is the default value for the k0s.inventory.yml template.
`need_transfer`	Indicates that KUMA components are moved in a Kubernetes cluster.	• `false` is the default value for the single.inventory.yml and distributed.inventory.yml templates. • `true` – is the default value for the k0s.inventory.yml template.
`airgap`	Indicates that there is no internet connection.	`true` – is the default value for the k0s.inventory.yml template.
`no_firewall_actions`	Indicates that the installer has completed the firewall configuration steps on the hosts.	• `true` – — when the installer is started, the firewall configuration steps on the hosts are not performed. • `false` — default value in all templates. The installer performs the

		firewall configuration steps on the hosts. If this setting is not specified in the template, the installer performs the firewall configuration steps on the hosts.
`generate_etc_hosts`	Indicates that the machines are registered in the DNS zone of your organization. In this case, the installer will automatically add the IP addresses of the machines from the inventory file to the /etc/hosts files on the machines where KUMA components are installed. The specified IP addresses must be unique.	• `false`. • `true`.
`deploy_example_services`	Indicates the creation of predefined services during installation.	• `false`: no services are needed. The default value for the distributed.inventory.yml and k0s.inventory.yml templates. • `true`: services must be created. The default value for the single.inventory.yml template.
`low_resources`	Indicates that KUMA is installed in environments with limited computing resources. In this case, the Core can be installed on a host that has 4 GB of free disk space. By default, there is no variable.	

kuma block

This block lists the settings of KUMA components deployed outside of the Kubernetes cluster.

The following sections are available in the block:

- In the `vars` section, you can specify the variables that are applied to all hosts indicated in the `kuma` block.
- In the `children` section you can list groups of component settings:
  - `kuma_core`—KUMA Core settings. This may contain only one host.
  - `kuma_collector`—settings of KUMA collectors. Can contain multiple hosts.
  - `kuma_correlator`—settings of KUMA correlators. Can contain multiple hosts.

- `kuma_storage`—settings of KUMA storage nodes. Can contain multiple hosts.

kuma_k0s block

This block defines the settings of the Kubernetes cluster that ensures high availability of KUMA. This block is only available in an inventory file that is based on k0s.inventory.yml.template.

Each host in this block must have its unique FQDN or IP address indicated in the ansible_host parameter, except for the host in the kuma_lb section which must have its FQDN indicated. Hosts must not be duplicated in groups.

For a demo installation, you may combine controller with a worker node. Such a configuration does not provide high availability for the Core and is only intended for demonstration of functionality or for testing the software environment.

The minimum configuration for high availability must include 3 dedicated controllers, 2 worker nodes, and 1 load balancer. For industrial operation, it is recommended to use dedicated worker nodes and controllers. If a cluster controller is under workload and the pod with the KUMA Core is hosted on the controller, disabling the controller will result in a complete loss of access to the Core.

The following sections are available in the block:

- In the `vars` section, you can specify the variables that are applied to all hosts indicated in the `kuma` block.
- The `children` section defines the settings of the Kubernetes cluster that ensures high availability of KUMA.

The table below shows a list of 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. The user installs the balancer on their own. If the `kuma_managed_lb = true` parameter is indicated within the group, the load balancer will be automatically configured during KUMA installation, the necessary network TCP ports will be opened on its host (6443, 8132, 9443, 7209, 7210, 7220, 7222, 7223), and a restart will be performed to apply the changes.	
`kuma_control_plane_master`	A host that acts as a dedicated primary controller for the cluster.	Groups for specifying the primary controller. A host must be assigned to only one of them.
`kuma_control_plane_master_worker`	A host that combines the role of the primary controller and a worker node 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_control_plane`	Hosts that act as a dedicated cluster controller.	Groups for specifying

`kuma_control_plane_worker`	Hosts that combine the role of controller and worker node 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"`.	secondary controllers.
`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"`.	

Installation on a single server

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

1. Ensure 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 program components in that file. The installer uses the single.inventory.yml file to deploy KUMA.
3. Install the program.

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

If necessary, you can move application 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 unpacked 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 name >.tar.gz` installer to the server and unpack 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 folder by executing the following command:

   ```
   cd kuma-ansible-installer
   ```
3. Copy the single.inventory.yml.template 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
   ```

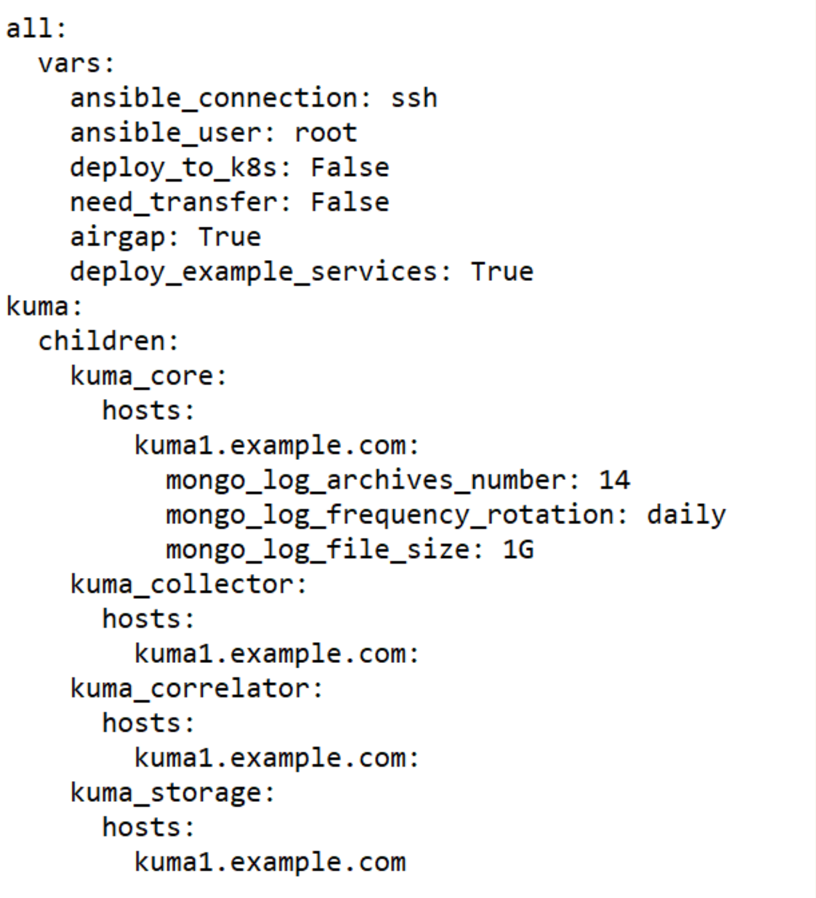
   The predefined services will appear only as a result of the initial installation of KUMA. If you are upgrading the system using the same inventory file, the predefined services are not re-created.

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 program. You can use it to add components to the system or remove KUMA.

Sample inventory file for installation on a single server



```
all:
  vars:
    ansible_connection: ssh
    ansible_user: root
    deploy_to_k8s: False
    need_transfer: False
    airgap: True
    deploy_example_services: True
kuma:
  children:
    kuma_core:
      hosts:
        kuma1.example.com:
          mongo_log_archives_number: 14
          mongo_log_frequency_rotation: daily
          mongo_log_file_size: 1G
    kuma_collector:
      hosts:
        kuma1.example.com:
    kuma_correlator:
      hosts:
        kuma1.example.com:
    kuma_storage:
      hosts:
        kuma1.example.com
```

Installing the program 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 unpacked into the kuma-ansibleinstaller directory.

2. Go to the directory with the unpacked installer.

3. Place the license 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
   ```

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

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

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

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

As a result, all KUMA components are installed. 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 the first login, change the password of the admin account

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

You can expand the installation to a distributed installation.

Distributed installation

Distributed installation of KUMA involves multiple steps:

1. Verifying that the hardware, software, and installation requirements for KUMA are satisfied.

2. Preparing the test machine.

   The test machine is used during the program installation process: the installer files are unpacked and run on it.

3. Preparing the target machines.

   The program components 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 program components. The installer uses this inventory file to deploy KUMA.

5. Installing the program.

Install the program 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, and agents.

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

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

Preparing the test machine

To prepare the test machine for the KUMA installation:

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

2. Generate an SSH key for authentication on the SSH servers of the target machines by executing the following command:

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

   If SSH root access is blocked on the test machine, generate an SSH key for authentication on the SSH servers of the target machines using **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
   > ```

   ```
   sudo ssh-keygen -f /home/<name of the user from 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 test machine has network access to all the target machines by host name and copy the SSH key to each target machine by carrying out the following command:

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

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

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

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

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

The test machine is ready for the KUMA installation.

Preparing the target machine

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

1. Ensure that hardware, software, and installation requirements are met.
2. Specify the host name. We recommend specifying the FQDN. For example, kuma1.example.com.

   You should 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 program components.
3. Register the target machine in your organization's DNS zone to allow host names to be translated 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 test 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:

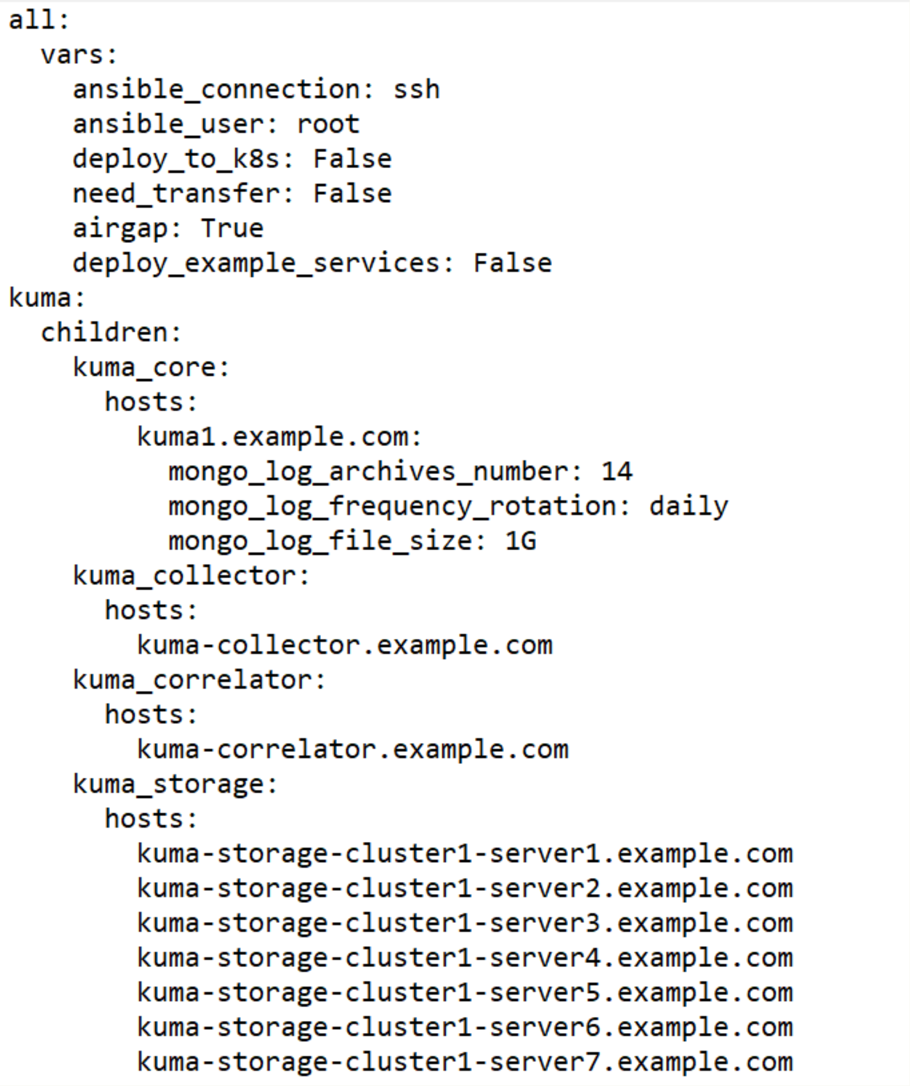
1. Go to the KUMA installer folder by executing the following command:

   ```
   cd kuma-ansible-installer
   ```
2. Copy the distributed.inventory.yml.template template and create an inventory file named distributed.inventory.yml:

   ```
   cp distributed.inventory.yml.template distributed.inventory.yml
   ```
3. Edit the settings in the distributed.inventory.yml inventory file.

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

Sample inventory file for distributed installation



```
all:
  vars:
    ansible_connection: ssh
    ansible_user: root
    deploy_to_k8s: False
    need_transfer: False
    airgap: True
    deploy_example_services: False
kuma:
  children:
    kuma_core:
      hosts:
        kuma1.example.com:
          mongo_log_archives_number: 14
          mongo_log_frequency_rotation: daily
          mongo_log_file_size: 1G
    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
```

Installing the program in a distributed configuration

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

To install KUMA:

1. On the test machine, open the folder containing the unpacked installer.

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

2. Place the license key file in the <installer directory>/roles/kuma/files/ directory.

   The key file must be named license.key.

3. Run the installer from the folder with the unpacked installer:

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

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

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

KUMA components are installed. The screen will display the URL of the KUMA web interface and the user name and password that must be used to access the web interface.

By default, the KUMA web interface address is `https://<FQDN or IP address of the core component>:7220`.

Default login credentials (after the first login, you must change the password of the admin account):
- user name — `admin`
- password— `mustB3Ch@ng3d!`

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

Modifying the self-signed web console certificate

> Before changing KUMA certificate, make sure to back up the previous certificate and key with the names external.cert.old and external.key.old respectively.

After installing the KUMA Core, the installer creates the following certificates in the /opt/kaspersky/kuma/core/certificates folder:

- Self-signed root certificate ca.cert with the ca.key.

  Signs all other certificates that are used for internal communication between KUMA components.

- The internal.cert certificate signed with the root certificate, and the Core server internal.key.

  Used for internal communication between KUMA components.

- KUMA web console external.cert certificate and external.key.

  Used in the KUMA web console and for REST API requests.

  You can use your company certificate and key instead of self-signed web console certificate. 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 KUMA web console certificate with an external certificate:

1. Switch to root user operation:

   ```
   sudo -i
   ```

2. Go to the certificates directory:

   ```
   cd /opt/kaspersky/kuma/core/certificates
   ```

3. Make a backup copy of the current certificate and key:

   ```
   mv external.cert external.cert.old && mv external.key external.key.old
   ```

4. In OpenSSL, convert the PFX file to a certificate and an 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.

5. Place the returned external.cert certificate and external.key files in the /opt/kaspersky/kuma/core/certificates directory.

6. Change the owner of the key files:

```
chown kuma:kuma external.cert external.key
```

7. Restart KUMA:

```
systemctl restart kuma-core
```

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

Your company certificate and key have been replaced.

Distributed installation in a high availability configuration

You can make KUMA high availability by deploying KUMA Core on a Kubernetes cluster and by using an external TCP traffic balancer.

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

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

- Install KUMA on a Kubernetes cluster from scratch.
- **Migrate the Core of an existing KUMA installation to the 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 3.0.2 or when a new installation was performed. In the inventory file, set the deploy_to_k8s, need_transfer and airgap parameters to true. The deploy_example_services parameter must be set to false.

   Sample inventory file with 3 dedicated controllers, 2 worker nodes, and 1 balancer.

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 the inventory file, it looks for an installed KUMA Core on all hosts where you want 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 the following systemd services from the host: kuma-core, kuma-mongodb, kuma-victoria-metrics, kuma-vmalert, and kuma-grafana.
- 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/log
  - /opt/kaspersky/kuma/core/logs
  - /opt/kaspersky/kuma/grafana/bin
  - /opt/kaspersky/kuma/mongodb/bin
  - /opt/kaspersky/kuma/mongodb/log
  - /opt/kaspersky/kuma/victoria-metrics/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.

- On the Core host, it migrates the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/grafana → /opt/kaspersky/kuma/grafana.moved
  - /opt/kaspersky/kuma/mongodb → /opt/kaspersky/kuma/mongodb.moved
  - /opt/kaspersky/kuma/victoria-metrics → /opt/kaspersky/kuma/victoria-metrics.moved

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

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 convert the names of the /opt/kaspersky/kuma/*.moved directories back to their original format.

If an /etc/hosts file with lines not related to addresses in the range 127.X.X.X was used on the Core host, 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.

Minimum configuration

There are 2 possible roles for nodes in Kubernetes:

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

To perform a high availability installation of KUMA, you will need:

- 3 dedicated controllers
- 2 worker nodes
- 1 balancer

To ensure the adequate performance of KUMA Core in Kubernetes, it is critically important to allocate 3 dedicated nodes with a single controller role. This will provide high availability for the Kubernetes cluster and will ensure that the workload (KUMA processes and other processes) cannot affect the tasks associated with managing the Kubernetes cluster. If you are using virtualization tools, make sure that the nodes are located on different physical servers and that these physical servers do not act as worker nodes.

For a demo installation of KUMA, you may combine the controller and working node 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 node role. KUMA cannot be upgraded to later versions if any of the nodes combine the controller and worker node roles.

Additional requirements for deploying KUMA Core in Kubernetes

To protect the KUMA network infrastructure using Kaspersky Endpoint Security for Linux, first install KUMA in a Kubernetes cluster and then deploy Kaspersky Endpoint Security for Linux.

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

- General application installation requirements.
- The hosts that are planned to be used for Kubernetes cluster nodes do not use IP addresses from the following Kubernetes blocks:
  - serviceCIDR: 10.96.0.0/12
  - podCIDR: 10.244.0.0/16

  The traffic to the proxy servers is also excluded for the addresses of these blocks.
- The nginx load balancer is installed and configured (more details about configuring nginx). For example, you can use the following command for installation:

  ```
  sudo yum install nginx
  ```

  If you want nginx to be configured automatically during the KUMA installation, install nginx and provide access to it via SSH 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.

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;
    }
}
```

- An access key from the device on which KUMA is installed is added to the load balancer server.
- The SELinux module is NOT enabled on the balancer server in the operating system.
- The tar, systemctl, setfacl packages are installed on the hosts.

During KUMA installation, the hosts are automatically checked to meet the following hardware requirements. If these conditions are not met, the installation is terminated.

For demonstration purposes, you can disable the check of these conditions during installation by specifying the `low_resources: true` variable in the inventory file.

- Number of CPU cores (threads) – 12 or more.
- RAM – 22,528 MB or more.
- Available disk space in the /opt/ section – 1,000 GB or more.
- For initial installation, the /var/lib/ section must have at least 32 GB of available space. If the cluster is already installed on this node, the size of the required available space is reduced by the size of the /var/lib/k0s directory.

Additional requirements for the application installation in the Astra Linux Special Edition operating system

- Installing a high availability configuration of KUMA is supported for the Astra Linux Special Edition RUSB.10015-01 operating system (2022-1011SE17MD, update 1.7.2.UU.1). Core version 5.15.0.33 or higher is required.
- The following packages are installed on the machines intended for deploying a Kubernetes cluster:
  - open-iscsi
  - wireguard
  - wireguard-tools

  The packages can be installed using the following command:

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

Additional requirements for the application installation in the Oracle Linux operating system

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

- iscsi-initiator-utils
- wireguard-tools

Before installing packages, you must add the EPEL repository as a source:

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

The packages can be installed using the following command:

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

Installing KUMA on a Kubernetes cluster from scratch

Distributed installation of KUMA involves multiple steps:

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

   The test machine is used during the program installation process: the installer files are unpacked and run on it.
3. Preparing the target machines.

   The program components 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 program components. The installer uses this inventory file to deploy KUMA.

5. Installing the program.

   Install the program 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, and agents.

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

If necessary, you use your company's certificate as the KUMA web console certificate.

Preparing the test machine

To prepare the test machine for the KUMA installation:

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

2. Generate an SSH key for authentication on the SSH servers of the target machines by executing the following command:

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

   If SSH root access is blocked on the test machine, generate an SSH key for authentication on the SSH servers of the target machines using **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
   > ```

   ```
   sudo ssh-keygen -f /home/<name of the user from 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 test machine has network access to all the target machines by host name and copy the SSH key to each target machine by carrying out the following command:

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

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

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

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

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

The test machine is ready for the KUMA installation.

Preparing the target machine

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

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

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

   You should 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 program 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 test 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.

   Sample inventory file for a demo installation with the Core in Kubernetes ?

```
all:
vars:
ansible_connection: ssh
ansible_user: root
deploy_to_k8s: true
need_transfer: false
generate_etc_hosts: false
airgap: true
deploy_example_services: true
kuma:
children:
kuma_core:
hosts:
kuma.example.com:
mongo_log_archives_number: 14
mongo_log_frequency_rotation: daily
mongo_log_file_size: 1G
kuma_collector:
hosts:
kuma.example.com:
kuma_correlator:
hosts:
kuma.example.com:
kuma_storage:
hosts:
kuma.example.com:
shard: 1
replica: 1
```

```
keeper: 1

kuma_k0s:

children:

kuma_control_plane_master_worker:

hosts:

kuma-cpw.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"
```

For a demo installation, specify deploy_example_services: true. KUMA will deploy demo services on the specified hosts and assign the shard, replica, and keeper roles to the specified host; configuring these roles in the KUMA web interface is not necessary for a demo installation.

Sample inventory file for a distributed installation in a high availability configuration with 3 controllers, 2 worker nodes, and 1 balancer

```
all:

vars:

ansible_connection: ssh

ansible_user: root

deploy_to_k8s: true

need_transfer: false

generate_etc_hosts: false

airgap: true

deploy_example_services: false

kuma:

children:

kuma_core:

hosts:

kuma-core.example.com:

mongo_log_archives_number: 14

mongo_log_frequency_rotation: daily

mongo_log_file_size: 1G

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"
```

For such a configuration, specify the need_transfer: false, airgap: true, deploy_example_services: false, and 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.

Sample 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

airgap: true

deploy_example_services: false

kuma:

children:

kuma_core:

hosts:

kuma-core.example.com:

mongo_log_archives_number: 14

mongo_log_frequency_rotation: daily

mongo_log_file_size: 1G

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 3.0.2 or when a new installation was performed. In the k0s.inventory.yml inventory file, set deploy_to_k8s: true, need_transfer: true, airgap: true, deploy_example_services: false.

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

Installing the program in a high availability configuration

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

To install KUMA:

1. On the test machine, open the folder containing the unpacked installer.

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

2. Place the license key file in the <installer directory>/roles/kuma/files/ directory.

   The key file must be named license.key.

3. Run the installer from the folder with the unpacked installer:

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

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

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

KUMA components are installed. The screen will display the URL of the KUMA web interface and the user name and password that must be used to access the web interface.

By default, the KUMA web interface address is `https://<FQDN or IP address of the core component>:7220`.

Default login credentials (after the first login, you must change the password of the admin account):
- user name — `admin`
- password— `mustB3Ch@ng3d!`

We recommend backing up the inventory file that you used to install the program. You can use it 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 3.0.2 or when a new installation was performed. In the inventory file, set the deploy_to_k8s, need_transfer and airgap parameters to true. The deploy_example_services parameter must be set to false.

Sample inventory file with 3 dedicated controllers, 2 worker nodes, and 1 balancer.

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 the inventory file, it looks for an installed KUMA Core on all hosts where you want 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 the following systemd services from the host: kuma-core, kuma-mongodb, kuma-victoria-metrics, kuma-vmalert, and kuma-grafana.
- 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/log
  - /opt/kaspersky/kuma/core/logs
  - /opt/kaspersky/kuma/grafana/bin
  - /opt/kaspersky/kuma/mongodb/bin
  - /opt/kaspersky/kuma/mongodb/log
  - /opt/kaspersky/kuma/victoria-metrics/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it migrates the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/grafana → /opt/kaspersky/kuma/grafana.moved

- /opt/kaspersky/kuma/mongodb → /opt/kaspersky/kuma/mongodb.moved
- /opt/kaspersky/kuma/victoria-metrics → /opt/kaspersky/kuma/victoria-metrics.moved

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

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 convert the names of the /opt/kaspersky/kuma/*.moved directories back to their original format.

If an /etc/hosts file with lines not related to addresses in the range 127.X.X.X was used on the Core host, 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.

KUMA Core availability under various scenarios

KUMA Core availability under 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 deployment is complete, which takes less than one minute, the KUMA web interface becomes available again via URLs that use 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. A restored node can participate in replication of a disk volume of the Core service.

- **Malfunction or network disconnection of a worker node containing a replica of the KUMA Core drive on which the Core service is not currently deployed.**

  Access to the KUMA web interface is not lost via URLs that use the FQDN of the load balancer. The network storage creates a replica of the running Core disk volume on other running nodes. When accessing KUMA via a URL with the FQDN of running nodes, there is no disruption.

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

  Worker nodes operate in normal mode. 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 machines 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

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 a 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 KUMA Core can be accessed at the URL `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. Access can be obtained through any worker node whose `extra_args` parameter contains the value `kaspersky.com/kuma-ingress=true`.

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 is UTC+0 in the `DeviceTimeZone` field.
- In generated reports, the user will see the difference between the time the report was generated and the time in the browser.
- In the dashboard, the user will see 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 within the Kubernetes cluster is displayed).

Managing KUMA web console certificates in a high availability configuration

Modifying the self-signed web console certificate

To replace the self-signed KUMA web console certificate with your corporate certificate:

1. Connect to the primary controller of the cluster via ssh:

   ```
   ssh <user name>@<FQDN of the primary controller>
   ```

2. Go to the home directory of the user or create a new directory for subsequent operations and go to it.

3. Back up the current certificate and key to the current directory on the cluster controller:

   ```
   export POD=$(k0s kubectl get pods --namespace kuma -l "app=core" -o jsonpath="{.items[0].metadata.name}")
   ```

   ```
   sudo k0s kubectl cp --no-preserve -c core kuma/$POD:/opt/kaspersky/kuma/core/certificates/external.cert ./external.cert.old
   ```

   ```
   sudo k0s kubectl cp --no-preserve -c core kuma/$POD:/opt/kaspersky/kuma/core/certificates/external.key ./external.key.old
   ```

4. Prepare the custom certificate and key that you want to use as replacement.

   In OpenSSL, convert the PFX file to a certificate and an encrypted key in PEM format:

   ```
   sudo openssl pkcs12 -in kumaWebIssuedByCorporateCA.pfx -nokeys -out external.cert
   ```

   ```
   sudo 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.

5. Place the obtained 'external.cert' certificate file and 'external.key' key file in the current directory on the cluster controller and then copy them to the file system of the KUMA Core pod:

   ```
   export POD=$(k0s kubectl get pods --namespace kuma -l "app=core" -o jsonpath="{.items[0].metadata.name}")
   ```

   ```
   sudo k0s kubectl cp --no-preserve ./external.cert kuma/$POD:/opt/kaspersky/kuma/core/certificates/external.cert -c core
   ```

   ```
   sudo k0s kubectl cp --no-preserve ./external.key kuma/$POD:/opt/kaspersky/kuma/core/certificates/external.key -c core
   ```

6. Restart the KUMA Core:

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

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

The self-signed certificate of the web console is replaced with your corporate certificate.

Rolling back the changes

To roll back the changes you made and return to using the previous certificate and key:

1. Go to the home directory of the user on the primary controller and run the following commands:

   ```
   sudo export POD=$(k0s kubectl get pods --namespace kuma -l "app=core" -o jsonpath="{.items[0].metadata.name}")
   ```

   ```
   sudo k0s kubectl cp --no-preserve ./external.cert.old kuma/$POD:/opt/kaspersky/kuma/core/certificates/external.cert -c core
   ```

   ```
   sudo k0s kubectl cp --no-preserve ./external.key.old kuma/$POD:/opt/kaspersky/kuma/core/certificates/external.key -c core
   ```

2. Restart the KUMA Core:

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

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

The changes are rolled back and the previous certificate and key of the web console are being used.

KUMA backup

KUMA allows you to back up the KUMA Core database and certificates. The backup function is intended for restoring KUMA. To move or copy the resources, use the resource export and import functions.

Backup can be done in the following ways:

- Using the REST API
- Using the /opt/kaspersky/kuma/kuma executable file

  > The KUMA backup using the kuma executable file is not available in KUMA higher than 2.1.

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.
- Backup of collectors is not required unless the collectors have an SQL connection. When restoring such collectors, you should revert to the original initial value of the ID.
- If KUMA cannot start after the restore, it is recommended to reset the kuma database in MongoDB.

  How to reset a database in MongoDB

If the KUMA Core fails to start after data recovery, the recovery must be performed again but this time the kuma database in MongoDB® must be reset.

To restore KUMA data and reset the MongoDB database:

1. Log in to the OS of the server where the KUMA Core is installed.
2. On the KUMA Core server, run the following command:

   ```
   sudo systemctl stop kuma-core
   ```

3. Log in to MongoDB by running the following commands:

   a. `cd /opt/kaspersky/kuma/mongodb/bin/`

   b. `./mongo`

4. Reset the MongoDB database by running the following commands:

   a. `use kuma`

   b. `db.dropDatabase()`

5. Log out of the MongoDB database by pressing **`Ctrl+C`**.
6. Restore data from a backup copy by running the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma tools restore --src <path to folder containing
   backup copy> --certificates
   ```

   The `--certificates` flag is optional and is used to restore certificates.

7. Start KUMA by running the following command:

   ```
   sudo systemctl start kuma-core
   ```

8. Rebuild the services using the recovered service resource sets.

Data is restored from the backup.

KUMA backup using the kuma file

To perform a backup:

1. Log in to the OS of the server where the KUMA Core is installed.
2. Execute the following command of the kuma executable file:

   ```
   sudo /opt/kaspersky/kuma/kuma tools backup --dst <path to folder for backup copy> --
   certificates
   ```

The backup copy has been created.

To restore data from a backup:

1. Log in to the OS of the server where the KUMA Core is installed.

2. On the KUMA Core server, run the following command:

   ```
   sudo systemctl stop kuma-core
   ```

3. Execute the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma tools restore --src <path to folder containing backup
   copy> --certificates
   ```

4. Start KUMA by running the following command:

   ```
   sudo systemctl start kuma-core
   ```

5. In the KUMA web interface, in the **Resources** → **Active services** section, select all services and click the **Reset certificate** button.

6. Reinstall the services with the same ports and IDs.

Data is restored from the backup.

Modifying the configuration of KUMA

The following KUMA configuration changes can be performed.

- Extending an all-in-one installation to a distributed installation.

To expand an all-in-one installation to a distributed installation:

1. Create a backup copy of KUMA.
2. Remove the pre-installed correlator, collector, and storage services from the server.
   a. In the KUMA web interface, under **Resources** → **Active services**, select a service and click **Copy ID**. On the server where the services were installed, run the service removal command:

      ```
      sudo /opt/kaspersky/kuma/kuma <collector/correlator/storage> --id <service ID copied from the KUMA web interface> --uninstall
      ```

      Repeat the removal command for each service.

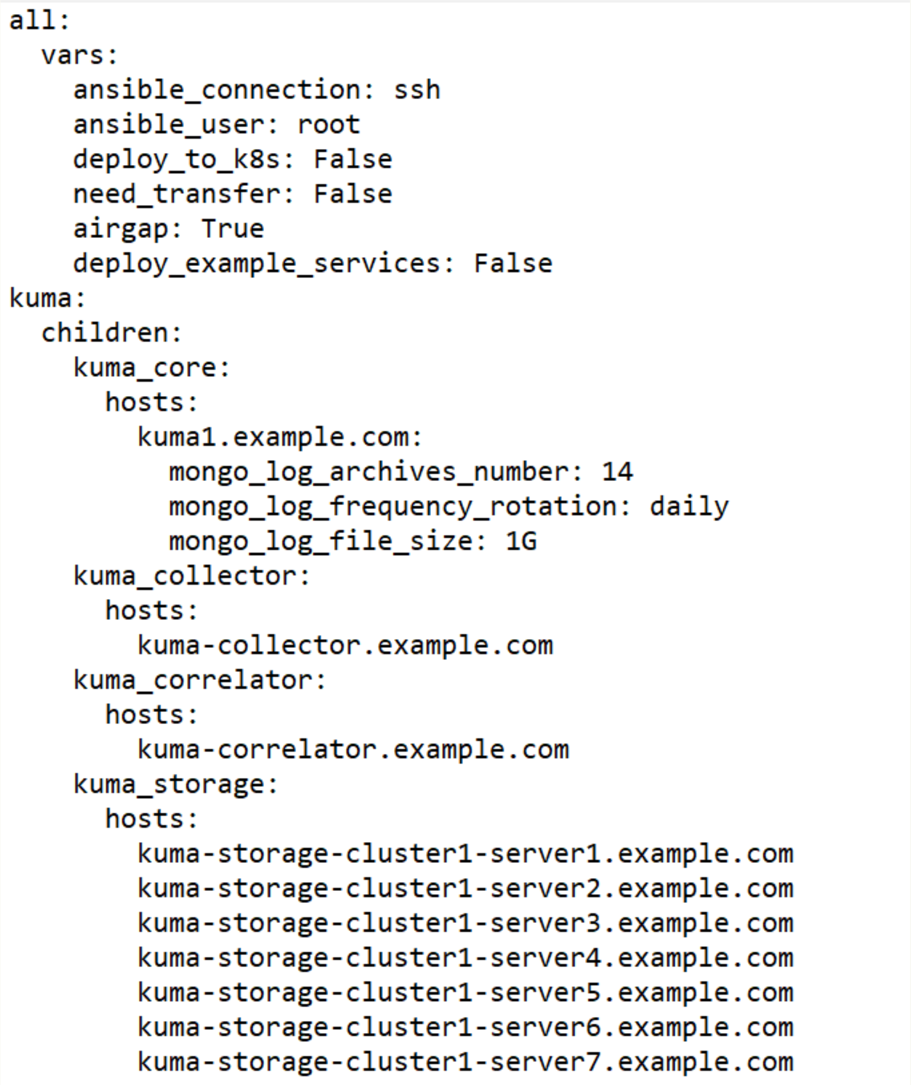
   b. Then remove the services in the KUMA web interface:

   As a result, only the KUMA Core remains on the initial installation server.

3. Prepare the distributed.inventory.yml inventory file and in that file, specify the initial all-in-one initial installation server in the `kuma_core` group.

   In this way, the KUMA Core remains on the original server, and you can deploy the other components on other servers. Specify the servers on which you want to install the KUMA components in the inventory file.

   Sample inventory file for expanding an all-in-one installation to a distributed installation



```
all:
  vars:
    ansible_connection: ssh
    ansible_user: root
    deploy_to_k8s: False
    need_transfer: False
    airgap: True
    deploy_example_services: False
kuma:
  children:
    kuma_core:
      hosts:
        kuma1.example.com:
          mongo_log_archives_number: 14
          mongo_log_frequency_rotation: daily
          mongo_log_file_size: 1G
    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
```

4. Create and install the storage, collector, correlator, and agent services on other machines.

   a. After you specify the settings for all sections in the distributed.inventory.yml inventory file, run the installer on the test machine.

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

      Running the command causes the files necessary to install the KUMA components (storages, collectors, correlators) to appear on each target machine specified in the distributed.inventory.yml inventory file.

   b. Create storage, collector, and correlator services.

The expansion of the installation is completed.

- **Adding servers for collectors to a distributed installation** ?.

The following instructions show how to add one or more servers to an existing infrastructure and 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. Ensure that the target machines meet hardware, software, and installation requirements.
2. On the test machine, go to the directory with the unpacked KUMA installer by running the following command:

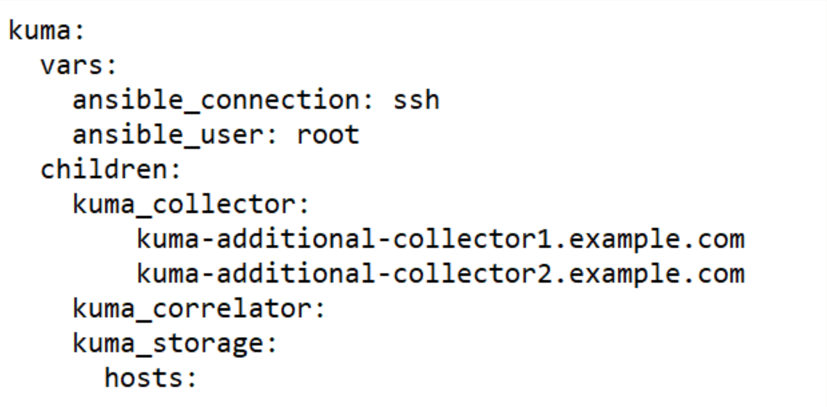
   ```
   cd kuma-ansible-installer
   ```

3. Copy the expand.inventory.yml.template template to create an inventory file called expand.inventory.yml:

   ```
   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.

   Sample expand.inventory.yml inventory file for adding collector servers

   ```
   kuma:
     vars:
       ansible_connection: ssh
       ansible_user: root
     children:
       kuma_collector:
           kuma-additional-collector1.example.com
           kuma-additional-collector2.example.com
       kuma_correlator:
       kuma_storage:
         hosts:
   ```

5. On the test machine, run the following command as root from the directory with the unpacked installer:

   ./expand.sh expand.inventory.yml

   Running this command on each target machine specified in the expand.inventory.yml inventory file creates files for creating and installing the collector.

6. Create and install the collectors. A KUMA collector consists of a client part and a server part, therefore creating a collector involves two steps.

   1. Creating the client part of the collector, which includes a set of resources and the collector service.

      To create a set of resources 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 and displayed on the screen. Copy the installation command and proceed to the next step.

   2. Creating the server part of the collector.

   a. On the target machine, 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 machine. You can check the status of the service in the web interface under **Resources** → **Active services**.

b. Run the same command on each target machine 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 ?.

The following instructions show how to add one or more servers to an existing infrastructure and then install correlators 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. Ensure that the target machines meet hardware, software, and installation requirements.

2. On the test machine, go to the directory with the unpacked KUMA installer by running the following command:

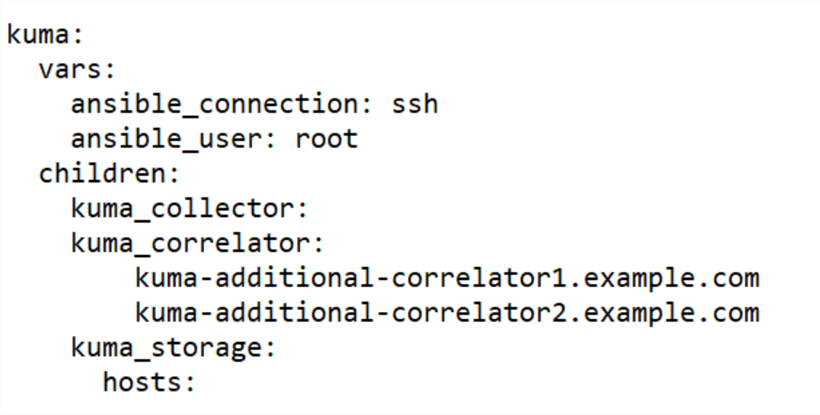
   ```
   cd kuma-ansible-installer
   ```

3. Copy the expand.inventory.yml.template template to create an inventory file called expand.inventory.yml:

   ```
   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.

   Sample expand.inventory.yml inventory file for adding correlator servers

   ```
   kuma:
     vars:
       ansible_connection: ssh
       ansible_user: root
     children:
       kuma_collector:
       kuma_correlator:
           kuma-additional-correlator1.example.com
           kuma-additional-correlator2.example.com
       kuma_storage:
         hosts:
   ```

5. On the test machine, run the following command as root from the directory with the unpacked installer:

   ```
   ./expand.sh expand.inventory.yml
   ```

   Running this command on each target machine specified in the expand.inventory.yml inventory file creates files for creating and installing the correlator.

6. Create and install the correlators. A KUMA correlator consists of a client part and a server part, therefore creating a correlator involves two steps.

   1. Creating the client part of the correlator, which includes a set of resources 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 and displayed on the screen. Copy the installation command and proceed to the next step.

   2. Creating the server part of the correlator.

   a. On the target machine, 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 machine. You can check the status of the service in the web interface under **Resources** → **Active services**.

b. Run the same command on each target machine 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 ?.

The following instructions show 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. Ensure that the target machines meet hardware, software, and installation requirements.

2. On the test machine, go to the directory with the unpacked KUMA installer by running the following command:

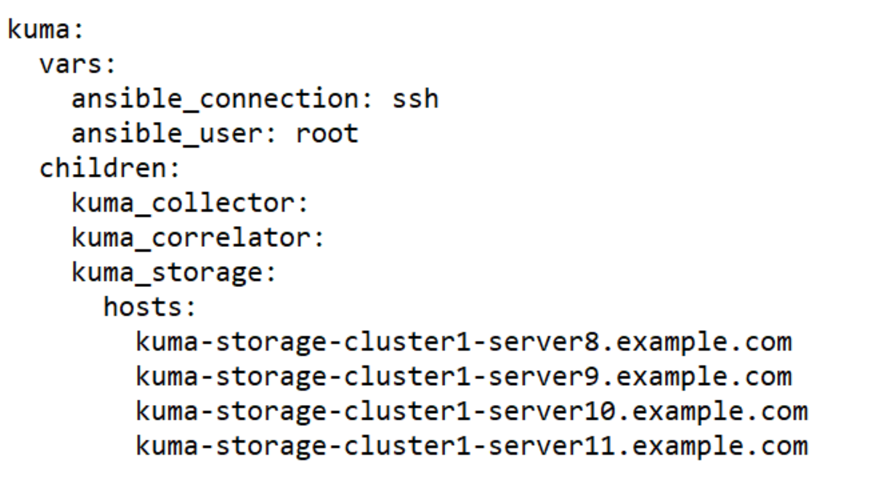
   ```
   cd kuma-ansible-installer
   ```

3. Copy the expand.inventory.yml.template template to create an inventory file called expand.inventory.yml:

   ```
   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 two shards, each of which contains two replicas. In the expand.inventory.yml inventory file, you must only specify the FQDN, the roles of shards and replicas are assigned later in the KUMA web interface by following the steps of the instructions. You can adapt this example to suit your needs.

   Sample 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
           kuma-storage-cluster1-server9.example.com
           kuma-storage-cluster1-server10.example.com
           kuma-storage-cluster1-server11.example.com
   ```

5. On the test machine, run the following command as root from the directory with the unpacked installer:

   ```
   ./expand.sh expand.inventory.yml
   ```

   ```
   Running this command on each target machine 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. You must edit the storage settings of the existing cluster:

   a. In the **Resources** → **Storages** section, select an existing storage and open the storage for editing.

   b. In the **ClickHouse cluster nodes** section, click **Add nodes** and specify roles in the fields for the new node. The following example shows how to specify identifiers to add two shards, containing two replicas each, to an existing cluster. You can adapt the example to suit your needs.

      Example:

      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. 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 storage you edited at the previous step and click **Create service**. Do the same for each ClickHouse storage node you are adding.

   As a result, the number of created services must be the same as the number of nodes added to the ClickHouse cluster, that is, four services for four nodes. The created storage services are displayed in the KUMA web interface in the **Resources** → **Active services** section. Now storage services must be installed on each server by using the service ID.

8. Now storage services must be installed 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 need it for running the service installation command.

   b. Compose and run the following command on the target machine:

   ```
   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 machine. 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 machine listed in the 'storage' section of the expand.inventory.yml inventory file, one machine at a time. On each machine, the unique service ID within the cluster must be specified in the installation command.

9. To apply changes to a running cluster, in the KUMA web interface, under **Resources** → **Active services**, select the check box 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 an additional storage cluster** ?.

The following instructions show how to add an additional storage cluster to existing infrastructure. You can use these instructions as an example and adapt them to your requirements.

To add an additional storage cluster:

1. Ensure that the target machines meet hardware, software, and installation requirements.

2. On the test machine, go to the directory with the unpacked KUMA installer by running the following command:

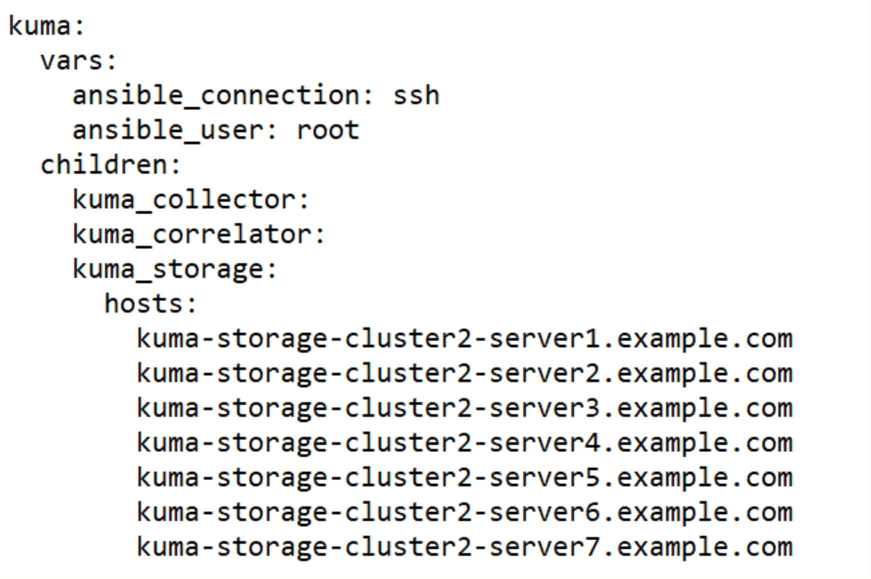
   ```
   cd kuma-ansible-installer
   ```

3. Copy the expand.inventory.yml.template template to create an inventory file called expand.inventory.yml:

   ```
   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 three dedicated keepers and two shards, each of which contains two replicas. In the expand.inventory.yml inventory file, you must only specify the FQDN, the roles of keepers, shards, and replicas are assigned later in the KUMA web interface by following the steps of the instructions. You can adapt this example to suit your needs.

   Sample expand.inventory.yml inventory file for adding an additional storage cluster

   ```
   kuma:
     vars:
       ansible_connection: ssh
       ansible_user: root
     children:
       kuma_collector:
       kuma_correlator:
       kuma_storage:
         hosts:
           kuma-storage-cluster2-server1.example.com
           kuma-storage-cluster2-server2.example.com
           kuma-storage-cluster2-server3.example.com
           kuma-storage-cluster2-server4.example.com
           kuma-storage-cluster2-server5.example.com
           kuma-storage-cluster2-server6.example.com
           kuma-storage-cluster2-server7.example.com
   ```

5. On the test machine, run the following command as root from the directory with the unpacked installer:

   ```
   ./expand.sh expand.inventory.yml
   ```

   Running this command on each target machine specified in the expand.inventory.yml inventory file creates files for creating and installing the storage.

6. Create and install a storage. For each storage cluster, you must create a separate storage, that is, three storages for three storage clusters. A storage consists of a client part and a server part, therefore creating a storage involves two steps.

   1. Creating the client part of the storage, which includes a set of resources and the storage service.

      a. 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 set of resources for a storage.

The created set of resources for the storage is displayed in the **Resources** → **Storages** section. Now you can create storage services for each ClickHouse cluster node.

b. 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 set of resources that you created for the storage at the previous step and click **Create service**. Do the same for each ClickHouse storage.

As a result, the number of created services must be the same as the number of nodes in the ClickHouse cluster, that is, fifty services for fifty nodes. The created storage services are displayed in the KUMA web interface in the **Resources** → **Active services** section. Now storage services must be installed to each node of the ClickHouse cluster by using the service ID.

2. Creating the server part of the storage.

a. On the target machine, create the server part of the storage: in the KUMA web interface, in the **Resources** → **Active services** section, select the relevant storage service and click **Copy ID**.

The service ID is copied to the clipboard; you need it for running the service installation command.

b. Compose and run the following command on the target machine:

```
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 machine. 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 machine listed in the 'storage' section of the expand.inventory.yml inventory file, one machine at a time. On each machine, the unique service ID within the cluster must be specified in the installation command.

d. Dedicated keepers are automatically started immediately after installation and are displayed in **the Resources** → **Active services** section with a 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 a red status. This is normal behavior for creating a new storage cluster or adding nodes to an existing storage cluster. As soon as the command to install services on all nodes of the cluster is executed, all services acquire 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.

An additional storage cluster is successfully added.

- Removing servers from a 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.
   a. Remove the server part of the service. Copy the service ID in the KUMA web interface and run the following command on the target machine:
      ```
      sudo /opt/kaspersky/kuma/kuma <collector/correlator/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
      ```
   b. Remove the client part of the service in the KUMA web interface in the **Active services** → **Delete** section.

      The service is removed.
2. Repeat step 1 for each server that you want to remove from the infrastructure.
3. Remove servers from the relevant sections of the distributed.inventory.yml inventory file to make sure the inventory file has up-to-date information in case of a KUMA update.

The servers are removed from the distributed installation.

- Removing a storage cluster from a distributed installation.

To remove one or more storage clusters from a distributed installation:

1. Remove the storage service on each cluster server that you want to removed from the distributed installation.
   a. Remove the server part of the storage service. Copy the service ID in the KUMA web interface and run 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.
2. 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 of a KUMA update or a configuration change.

The cluster is removed from the distributed installation.

- 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 3.0.2 or when a new installation was performed. In the inventory file, set the deploy_to_k8s, need_transfer and airgap parameters to true. The deploy_example_services parameter must be set to false.

   Sample inventory file with 3 dedicated controllers, 2 worker nodes, and 1 balancer.

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 the inventory file, it looks for an installed KUMA Core on all hosts where you want 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 the following systemd services from the host: kuma-core, kuma-mongodb, kuma-victoria-metrics, kuma-vmalert, and kuma-grafana.
- 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/log
  - /opt/kaspersky/kuma/core/logs
  - /opt/kaspersky/kuma/grafana/bin
  - /opt/kaspersky/kuma/mongodb/bin
  - /opt/kaspersky/kuma/mongodb/log
  - /opt/kaspersky/kuma/victoria-metrics/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.

- On the Core host, it migrates the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/grafana → /opt/kaspersky/kuma/grafana.moved
  - /opt/kaspersky/kuma/mongodb → /opt/kaspersky/kuma/mongodb.moved
  - /opt/kaspersky/kuma/victoria-metrics → /opt/kaspersky/kuma/victoria-metrics.moved

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

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 convert the names of the /opt/kaspersky/kuma/*.moved directories back to their original format.

If an /etc/hosts file with lines not related to addresses in the range 127.X.X.X was used on the Core host, 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.

Updating previous versions of KUMA

The update is performed the same way on all hosts using the installer and inventory file.

Version upgrade scheme:

2.0.x → 2.1.3 → 3.0.2

2.1.x → 2.1.3 → 3.0.2

2.1.3 → 3.0.2

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 update.

Preliminary steps

1. Creating a backup copy of the KUMA Core.
2. Make sure that all application installation requirements are met.
3. Make sure that MongoDB versions are compatible by running the following sequence of commands on the device where KUMA Core is located:

   ```
   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 update, ensure network accessibility of TCP port 7220 on the KUMA Core for 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 performing the update:
   - Use the existing storage of the cluster to create a storage service for the keeper in the web interface.
   - Install the service on a device with a 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 and configuration files of the service 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 (they will be subsequently 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 update, alert segmentation rules are not

migrated automatically.

8. To perform an update, you need a valid password from the admin user. If you forgot the admin user password, contact Technical Support to reset the current password and use the new password to perform the update at the next step.

Updating KUMA

1. Depending on the KUMA deployment scheme being used, do one the following:

- Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the application.
- 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 fill in the corresponding settings. To view a list of hosts and host roles in the current KUMA system, in the web interface, go to **Resources** → **Active services** section.

  The upgrade process completely reproduces 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** .

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 3.0.2 or when a new installation was performed. In the inventory file, set the deploy_to_k8s, need_transfer and airgap parameters to true. The deploy_example_services parameter must be set to false.

   Sample inventory file with 3 dedicated controllers, 2 worker nodes, and 1 balancer.

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 the inventory file, it looks for an installed KUMA Core on all hosts where you want 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 the following systemd services from the host: kuma-core, kuma-mongodb, kuma-victoria-metrics, kuma-vmalert, and kuma-grafana.
- 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/log
  - /opt/kaspersky/kuma/core/logs
  - /opt/kaspersky/kuma/grafana/bin
  - /opt/kaspersky/kuma/mongodb/bin
  - /opt/kaspersky/kuma/mongodb/log
  - /opt/kaspersky/kuma/victoria-metrics/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.

- On the Core host, it migrates the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/grafana → /opt/kaspersky/kuma/grafana.moved
  - /opt/kaspersky/kuma/mongodb → /opt/kaspersky/kuma/mongodb.moved
  - /opt/kaspersky/kuma/victoria-metrics → /opt/kaspersky/kuma/victoria-metrics.moved

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

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 convert the names of the /opt/kaspersky/kuma/*.moved directories back to their original format.

If an /etc/hosts file with lines not related to addresses in the range 127.X.X.X was used on the Core host, 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.

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 an error because KUMA could not start the Core service due to a timeout error and resource limit. 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 update the KUMA agents.

KUMA update completed successfully.

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 update.

Preliminary steps

1. Creating a backup copy of the KUMA Core.
2. Make sure that all application 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 admin user password, contact Technical Support to reset the current password and use the new password to perform the update at the next step.

Updating KUMA

1. Depending on the KUMA deployment scheme being used, do one the following:

- Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the application.
- 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 fill in the corresponding settings. To view a list of hosts and host roles in the current KUMA system, in the web interface, go to **Resources** → **Active services** section.

  The upgrade process completely reproduces 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**
  .

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 3.0.2 or when a new installation was performed. In the inventory file, set the deploy_to_k8s, need_transfer and airgap parameters to true. The deploy_example_services parameter must be set to false.

   Sample inventory file with 3 dedicated controllers, 2 worker nodes, and 1 balancer.

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 the inventory file, it looks for an installed KUMA Core on all hosts where you want 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 the following systemd services from the host: kuma-core, kuma-mongodb, kuma-victoria-metrics, kuma-vmalert, and kuma-grafana.
- 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/log
  - /opt/kaspersky/kuma/core/logs
  - /opt/kaspersky/kuma/grafana/bin
  - /opt/kaspersky/kuma/mongodb/bin
  - /opt/kaspersky/kuma/mongodb/log
  - /opt/kaspersky/kuma/victoria-metrics/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.

- On the Core host, it migrates the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/grafana → /opt/kaspersky/kuma/grafana.moved
  - /opt/kaspersky/kuma/mongodb → /opt/kaspersky/kuma/mongodb.moved
  - /opt/kaspersky/kuma/victoria-metrics → /opt/kaspersky/kuma/victoria-metrics.moved

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

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 convert the names of the /opt/kaspersky/kuma/*.moved directories back to their original format.

If an /etc/hosts file with lines not related to addresses in the range 127.X.X.X was used on the Core host, 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.

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 an error because KUMA could not start the Core service due to a timeout error and resource limit. 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 3.0.2

To install KUMA version 3.0.2 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 can restore data from backup.
2. Make sure that all application 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 being used, do one the following:

- Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the application.
- 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 fill in the corresponding settings. To view a list of hosts and host roles in the current KUMA system, in the web interface, go to **Resources** → **Active services** section.

The upgrade process completely reproduces 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**.

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 3.0.2 or when a new installation was performed. In the inventory file, set the deploy_to_k8s, need_transfer and airgap parameters to true. The deploy_example_services parameter must be set to false.

   Sample inventory file with 3 dedicated controllers, 2 worker nodes, and 1 balancer.

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 the inventory file, it looks for an installed KUMA Core on all hosts where you want 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 the following systemd services from the host: kuma-core, kuma-mongodb, kuma-victoria-metrics, kuma-vmalert, and kuma-grafana.
- 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/log
  - /opt/kaspersky/kuma/core/logs
  - /opt/kaspersky/kuma/grafana/bin
  - /opt/kaspersky/kuma/mongodb/bin
  - /opt/kaspersky/kuma/mongodb/log
  - /opt/kaspersky/kuma/victoria-metrics/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.

- On the Core host, it migrates the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/grafana → /opt/kaspersky/kuma/grafana.moved
  - /opt/kaspersky/kuma/mongodb → /opt/kaspersky/kuma/mongodb.moved
  - /opt/kaspersky/kuma/victoria-metrics → /opt/kaspersky/kuma/victoria-metrics.moved

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

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 convert the names of the /opt/kaspersky/kuma/*.moved directories back to their original format.

If an /etc/hosts file with lines not related to addresses in the range 127.X.X.X was used on the Core host, 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, upgrading from 2.1.3 to 3.0.2 does not update the universal dashboard layout.

   Possible solution: restart the Core service (kuma-core.service), and the data refresh interval configured for the layout will be used.

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 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.

Follow these steps to resolve the timeout error and successfully complete the update:

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 executing the commands and successfully starting the service in the second terminal, enter the administrator password again in the original first terminal when the installer prompts 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 the commands are executed, the update will be completed.

- 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 ClickHouse cluster fields, and click **Save**.

2. Restart the storage service.

The storage service will start with the specified parameters and will show a green status.

- DB::Exception error

  After updating KUMA, the storage may be in a red status, and its logs may show errors about suspicious strings.

  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.

Delete KUMA

To remove KUMA, use the Ansible tool and the user-generated inventory file.

To remove KUMA:

1. On the test machine, go to the installer folder:

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

2. Execute the following command:

   ```
   sudo ./uninstall.sh <inventory file>
   ```

   KUMA and all of the program data will be removed from the server.

The databases that were used by KUMA (for example, the ClickHouse storage database) and the information they contain must be deleted separately.

Special considerations for removing a high availability configuration of KUMA

The composition of the removed components depends on the value of the `deploy_to_k8s` parameter in the inventory file used to remove KUMA:

- `true` – the Kubernetes cluster created during the KUMA installation is deleted.
- `false` – all KUMA components except for the Core are deleted 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/"), it is recommended to restart the host the error is related to and try to uninstall 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 under **Settings** → **Tenants** in the KUMA web interface. You can sort the table by clicking on columns.

Available columns:

- **Name**—tenant name. The table can be filtered by this column.
- **EPS limit**—quota size for EPS (events processed per second) allocated to the tenant out of the overall EPS quota determined by the license.
- **Description**—description of the tenant.
- **Disabled**—indicates that the tenant is inactive.

  By default, inactive tenants are not displayed in the table. You can view them by selecting the **Show disabled** check box.
- **Created**—tenant creation date.

To create a tenant:

1. In the KUMA web interface under **Settings** → **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. The cumulative EPS of all tenants cannot exceed the EPS of the license.

4. If necessary, add a **Description** of the tenant. The description can contain no more than 256 Unicode characters.

5. Click **Save**.

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** → **Tenants** section of the KUMA web interface, select the relevant tenant by selecting the check box next to it, then select **Delete** in the toolbar.

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 a tenant is deleted, its services are automatically stopped, except the agents; events are no longer received or processed for the tenant; and the EPS of the tenant is no longer counted towards the cumulative EPS of the license. You can stop the Windows agent services manually in the Start → Services section, and to stop the Linux agent services, you can press Ctrl+C in the terminal in which the agent was started.

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 is read only.

You can create or edit user accounts under **Settings** → **Users** in the KUMA web interface. Users are also created automatically in the program 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.

The table of user accounts is displayed in the **Users** window of the KUMA web interface. You can use the **Search** field to look for users. You can sort the table based on the **User information** column by clicking the column header and selecting **Ascending** or **Descending**.

User accounts can be created, edited, or disabled. When editing user accounts (your own or the accounts of others), you can generate an API token for them.

By default, disabled user accounts are not displayed in the users table. However, they can be viewed by clicking the **User information** column and selecting the **Disabled users** check box.

To disable a user:

In the KUMA web interface, under **Settings** → **Users**, select the check box next to the relevant user and click **Disable user**.

User roles

KUMA users may have the following roles:

- *General administrator*—this role is designed for users who are responsible for the core functionality of KUMA systems. For example, they install system components, perform maintenance, work with services, create backups, and add users to the system. These users have full access to KUMA.
- *Tenant administrator*—this role is for users responsible for the core functionality of KUMA systems 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 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. 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 see resources of the shared tenant through the REST API.
- *Access to shared resources*—this role is intended for managing the shared tenant. Users with this role have read access to shared resources. Only a user with the General administrator role can edit resources of a shared tenant.
- *Interaction with NCIRCC*—this role can be selected if the license includes the NCIRCC module. Users with this role receive notifications by default.
- *Access to CII*—this role can be selected if the license includes the NCIRCC module. Users with this role receive notifications by default.

User roles rights

Web interface section and actions	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Access to shared resources	Access to NCIRCC	A t
Reports								
Create report template	filled in	filled in	filled in	filled in	no	no	no	n
View and edit templates and reports	filled in	filled in	filled in	filled in	no	no	no	n

Generate reports	filled in	filled in	filled in	filled in	no	no	no	n
Export generated reports	filled in	filled in	filled in	filled in	no	no	no	n
Delete templates and generated reports	filled in	filled in	filled in	filled in	no	no	no	n

Edit the settings for generating reports	filled in	filled in	filled in	filled in	no	no	no	n
Duplicate report template	filled in	filled in	filled in	filled in	no	no	no	n
Open the generated report by email	filled in	filled in	filled in	filled in	filled in	no	no	n
Dashboard								
View data on the dashboard and change layouts	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
View the Universal layout	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Add layouts	filled in	filled in	filled in	filled in	no	no	no	n

Edit and rename layouts	filled in	filled in	filled in	filled in	no	no	no	n
Delete layouts	filled in	filled in	filled in	filled in	no	no	no	n
Enable and disable the TV mode	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Resources → Services and Resources → Services → Active services								
View the list of active services	filled in	filled in	filled in	filled in	no	no	no	n
View the contents of the active list	filled in	filled in	filled in	filled in	no	no	no	n
Import/export/clear the contents of the	filled in	filled in	filled in	filled in	no	no	no	n

active list								
Create a set of resources for services	filled in	filled in	filled in	no	no	no	no	no
Create a service under Resources → Services → Active services	filled in	filled in	no	no	no	no	no	no
Delete services	filled in	filled in	no	no	no	no	no	no
Restart services	filled in	filled in	no	no	no	no	no	no
Update the settings of services	filled in	filled in	filled in	no	no	no	no	no
Reset certificates	filled in	filled in	no	no	no	no	no	no
Resources → Resources								
View the list of resources	filled in	filled in	filled in	filled in	no	filled in	no	no
Add resources	filled in	filled in	filled in	filled in	no	no	no	no
Duplicate resources	filled in	filled in	filled in	filled in	no	no	no	no
Edit resources	filled in	filled in	filled in	filled in	no	no	no	no
Delete resources	filled in	filled in	filled in	filled in	no	no	no	no

Import resources	filled in	filled in	filled in	filled in	no	no	no	n
View the repository, import the resources from the repository	filled in	filled in	filled in	no	no	no	no	n
Export resources	filled in	filled in	filled in	filled in	no	filled in	no	n
Source status → List of event sources								
View sources of events	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Change sources of events	filled in	filled in	filled in	no	no	no	no	n
Delete sources of events	filled in	filled in	filled in	no	no	no	no	n
Source status → Monitoring policies								
View monitoring policies	filled in	filled in	filled in	filled in	filled in	filled in	filled in	fil
Create monitoring policies	filled in	filled in	filled in	no	no	no	no	n
Edit monitoring policies	filled in	filled in	filled in	no	no	no	no	n
Delete monitoring policies	filled in	filled in	filled in	no	no	no	no	n
Assets								
View assets and asset categories	filled in	filled in	filled in	filled in	filled in	filled in	filled in	fil
Add/edit/delete asset categories	filled in	filled in	filled in	filled in	no	no	no	n
Add asset categories in a shared tenant	filled in	no	no	no	no	no	no	n

Link assets to an asset category of the shared tenant	filled in	filled in	filled in	filled in	no	no	no	n
Add assets	filled in	filled in	filled in	filled in	no	no	no	n
Edit assets	filled in	filled in	filled in	filled in	no	no	no	n
Delete assets	filled in	filled in	filled in	filled in	no	no	no	n
Import assets from Kaspersky Security Center	filled in	filled in	filled in	filled in	no	no	no	n
Start tasks on assets in Kaspersky Security Center	filled in	filled in	filled in	filled in	no	no	no	n
Run tasks on assets in Kaspersky Endpoint Detection and Response	filled in	filled in	filled in	filled in	no	no	no	n
Confirm updates to fix the asset vulnerabilities and accept the licensing agreements	filled in	filled in	no	no	no	no	no	n
Editing CII categorization in the asset card	filled in	no	no	no	no	no	no	fil
Editing custom fields of the assets (Settings → Assets)	filled in	filled in	filled in	filled in	no	no	no	n
Alerts								
View the list of alerts	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Change the severity of alerts	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Open the details of alerts	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Assign responsible users	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Close alerts	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Add comments to alerts	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Attach an event to alerts	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Detach an event from alerts	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Edit and delete someone else's filters	filled in	filled in	no	no	no	no	no	n

Incidents								
View the list of incidents	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Create blank incidents	filled in	filled in	filled in	filled in	filled in	no		
Manually create incidents from alerts	filled in	filled in	filled in	filled in	filled in	no		
Change the severity of incidents	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Open the incident details	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Assign executors	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Close incidents	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Add comments to incidents	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Attach alerts to incidents	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Detach alerts from incidents	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Edit and delete someone else's filters	filled in	filled in	no	no	no	no	no	n
Export incidents to NCIRCC	filled in	no	no	no	no	no	filled in	n
Send files to NCIRCC	filled in	no	no	no	no	no	filled in	n
Download files sent to NCIRCC	filled in	no	no	no	no	no	filled in	n
Export additional incident data to NCIRCC upon request	filled in	no	no	no	no	no	filled in	n
Send messages to NCIRCC	filled in	no	no	no	no	no	filled in	n
View messages from NCIRCC	filled in	no	no	no	no	no	filled in	n
View incident data exported to	filled in	no	no	no	no	no	filled in	n

NCIRCC								
Events								
View the list of events	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Search events	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Open the details of events	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Open statistics	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Perform a retroscan	filled in	filled in	filled in	no	no	no	no	n
Export events to a TSV file	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Edit and delete someone else's filters	filled in	filled in	no	no	no	no	no	n
Start ktl enrichment	filled in	filled in	filled in	filled in	no	no	no	n
Run tasks on Kaspersky Endpoint Detection and Response assets in event details	filled in	filled in	filled in	filled in	no	no	no	n
Create presets	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Delete presets	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
View and use presets	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Settings → Users								
View the list of users	filled in	no	no	no	no	no	no	n
Add a user	filled in	no	no	no	no	no	no	n
Edit a user	filled in	no	no	no	no	no	no	n
Generate token	filled in	filled in	filled in	filled in	filled in	filled in	filled in	fil
Change access rights for a token	filled in	filled in	filled in	filled in	filled in	filled in	filled in	fil

View the data of their own profile	filled in	filled in	filled in	filled in	filled in	filled in	filled in	fil
Edit the data of their own profile	filled in	filled in	filled in	filled in	filled in	filled in	filled in	fil
Settings → LDAP server								
View the LDAP connection settings	filled in	filled in	filled in	filled in	no	no	no	n
Edit the LDAP connection settings	filled in	filled in	no	no	no	no	no	n
Delete the configuration of an entire tenant from the settings	filled in	filled in	no	no	no	no	no	n
Import assets	filled in	filled in	no	no	no	no	no	n
Settings → Tenants								
View the list of tenants	filled in	no	no	no	no	no	no	n
Add tenants	filled in	no	no	no	no	no	no	n
Change tenants	filled in	no	no	no	no	no	no	n
Disable tenants	filled in	no	no	no	no	no	no	n
Settings → Domain authorization								
View the Active Directory connection settings	filled in	no	no	no	no	no	no	n
Edit the Active Directory connection settings	filled in	no	no	no	no	no	no	n
Add filters based on roles for tenants	filled in	no	no	no	no	no	no	n
Run tasks in Active Directory	filled in	filled in	filled in	no	no	no	no	n
Settings → General								
View the SMTP connection settings	filled in	no	no	no	no	no	no	n
Edit the SMTP	filled in	no	no	no	no	no	no	n

connection settings								
Settings → License								
View the list of added license keys	filled in	no	no	no	no	no	no	n
Add license keys	filled in	no	no	no	no	no	no	n
Delete license keys	filled in	no	no	no	no	no	no	n
Settings → Kaspersky Security Center								
View the list of successfully integrated Kaspersky Security Center servers	filled in	filled in	filled in	filled in	no	no	no	n
Add Kaspersky Security Center connections	filled in	filled in	no	no	no	no	no	n
Delete Kaspersky Security Center connections	filled in	filled in	no	no	no	no	no	n
Delete the configuration of an entire tenant from the settings	filled in	filled in	no	no	no	no	no	n
Start the tasks for importing Kaspersky Security Center assets	filled in	filled in	no	no	no	no	no	n
Settings → Kaspersky Industrial CyberSecurity for Networks								
View a list of KICS for Networks servers with which integration has been configured	filled in	filled in	no	no	no	no	no	n
Add and modify the settings of KICS for Networks integration	filled in	filled in	no	no	no	no	no	n
Delete the settings of KICS for Networks integration	filled in	filled in	no	no	no	no	no	n
Run the tasks to import assets from	filled in	filled in	no	no	no	no	no	n

the KICS for Networks settings								
Settings → Kaspersky Automated Security Awareness Platform								
View the ASAP integration settings	filled in	no	no	no	no	no	no	n
Edit the ASAP integration settings	filled in	no	no	no	no	no	no	n
Settings → Kaspersky Endpoint Detection and Response								
View the connection settings	filled in	filled in	filled in	filled in	no	no	no	n
Add, edit and disconnect the connections when the distributed solution mode is enabled	filled in	no	no	no	no	no	no	n
Enable the distributed solution mode	filled in	no	no	no	no	no	no	n
Add connections when the distributed solution mode is disabled	filled in	filled in	no	no	no	no	no	n
Delete the connections when the distributed solution mode is disabled	filled in	filled in	no	no	no	no	no	n
Delete the configuration of an entire tenant from the settings	filled in	filled in	no	no	no	no	no	n
Settings → Kaspersky CyberTrace								
View the CyberTrace integration settings	filled in	no	no	no	no	no	no	n
Edit the CyberTrace integration settings	filled in	no	no	no	no	no	no	n
Settings → IRP / SOAR								

View the settings for integration with IRP / SOAR	filled in	no	no	no	no	no	no	n
Edit the IRP/SOAR integration settings	filled in	no	no	no	no	no	no	n
Settings → Kaspersky Threat Lookup								
View the Threat Lookup integration settings	filled in	no	no	no	no	no	no	n
Edit the Threat Lookup integration settings	filled in	no	no	no	no	no	no	n
Settings → Alerts								
View the parameters	filled in	filled in	filled in	filled in	no	no	no	n
Edit the parameters	filled in	filled in	filled in	no	no	no	no	n
Delete the configuration of an entire tenant from the settings	filled in	filled in	filled in	no	no	no	no	n
Settings → Incidents → Automatic linking of alerts to incidents								
View the parameters	filled in	filled in	filled in	filled in	no	no	no	n
Edit the parameters	filled in	no	no	no	no	no	no	n
Settings → Incidents → Incident types								
View the categories reference	filled in	filled in	filled in	filled in	no	no	no	n
View the categories charts	filled in	filled in	filled in	filled in	no	no	no	n
Add categories	filled in	filled in	no	no	no	no	no	n
Edit categories	filled in	filled in	no	no	no	no	no	n
Delete categories	filled in	filled in	no	no	no	no	no	n
Settings → NCIRCC								
View the parameters	filled in	no	no	no	no	no	no	n
Edit the parameters	filled in	no	no	no	no	no	no	n
Settings → Hierarchy								

View the parameters	filled in	no	no	no	no	no	no	no
Edit the parameters	filled in	no	no	no	no	no	no	no
View incidents from child nodes	filled in	filled in	filled in	no	filled in	no	no	no
Settings → Asset audit								
Create, clone and edit the settings	filled in	filled in	filled in	no	no	no	no	no
View the parameters	filled in	filled in	filled in	filled in	no	no	no	no
Delete settings	filled in	filled in	filled in	no	no	no	no	no
Settings → Repository update								
View the parameters	filled in	filled in	filled in	no	no	no	no	no
Edit the parameters	filled in	no	no	no	no	no	no	no
Start the repository update task manually	filled in	filled in	filled in	no	no	no	no	no
Settings → Assets								
Add, edit, and delete the asset fields	filled in	no	no	no	no	no	no	no
Metrics								
Open metrics	filled in	no	no	no	no	no	no	no
Task manager								
View a list of your own tasks	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Finish your own	filled in	filled in	filled in	filled in	filled in	no	filled in	fil

tasks								
Restart your own tasks	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
View a list of all tasks	filled in	no	no	no	no	no	no	n
Finish any task	filled in	no	no	no	no	no	no	n
Restart any task	filled in	no	no	no	no	no	no	n
CyberTrace								
Open the section	filled in	no	no	no	no	no	no	n
Access to the data of tenants								
Access to tenants	filled in	filled in	filled in	filled in	filled in	no	filled in	fil
Shared tenant	filled in	filled in	filled in	filled in	filled in	filled in	filled in	fil

Main tenant	filled in	filled in	filled in	filled in	filled in	no	filled in	fil

Creating a user

To create a user account:

1. In the KUMA web interface, open **Settings** → **Users**.

   In the right part of the **Settings** section the **Users** table will be displayed.

2. Click the **Add user** button and set the parameters as described below.

   - **Name** (required)—enter the user name. The length of the string must be 1 to 128 Unicode characters.
   - **Login**(required) – enter a unique user name for the user account. Must contain from 3 to 64 characters (only a–z, A–Z, 0–9, . \ - _).
   - **Email** (required)—enter the unique email address of the user. Must be a valid email address.
   - **New password** (required)—enter the password to the user account. Password requirements:
     - 8 to 128 characters long.
     - At least one lowercase character.
     - At least one uppercase character.
     - At lease one numeral.
     - At least one of the following special characters: !, @, #, %, ^, &, *.
   - **Confirm password** (required)—enter the password again for confirmation.
   - **Disabled**—select this check box if you want to disable a user account. By default, this check box is cleared.

- In the **Tenants for roles** settings block, use the **Add field** buttons to specify which roles the user will perform on which tenants. A user can be assigned different roles in different tenants; multiple roles can be assigned within the same tenant.

3. Select or clear the check boxes that control access rights and user capabilities:

   - **Receive email notifications**—select this check box if you want the user to receive SMTP notifications from KUMA.

   - **Display non-printable characters**—select this check box if you want the KUMA web interface to display non-printing characters such as spaces, tab characters, and line breaks. If the **Display non-printable characters** check box is selected, you can press Ctrl/Command+* to enable and disable the display of non-printing characters.

     Spaces and tab characters are displayed in all input fields (except **Description**), in normalizers, correlation rules, filters and connectors, and in SQL queries for searching events in the **Events** section. Spaces are displayed as dots. A tab character is displayed as a dash in normalizers, correlation rules, filters and connectors. In other fields, a tab character is displayed as one or two dots.

     Line break characters are displayed in all input fields that support multi-line input, such as the event search field.

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. Click **Save**.

The user account will be created and displayed in the **Users** table.

Editing user

To edit 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 change the necessary settings in the user details area that opens on the right.

   - **Name** (required)—edit the user name. The length of the string must be 1 to 128 Unicode characters.

   - **Login**(required) – enter a unique user name for the user account. Must contain from 3 to 64 characters (only a–z, A–Z, 0–9, . \ - _).

   - **Email** (required)—enter the unique email address of the user. Must be a valid email address.

   - **Disabled**—select this check box if you want to disable a user account. By default, this check box is cleared.

   - In the **Tenants for roles** settings block, use the **Add field** buttons to specify which roles the user will perform on which 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, Access to shared resources), including management of additional role assignment to tenants.

3. Select or clear the check boxes that control access rights and user capabilities:
   - **Receive email notifications**—select this check box if you want the user to receive SMTP notifications from KUMA.
   - **Display non-printable characters**—select this check box if you want the KUMA web interface to display non-printing characters such as spaces, tab characters, and line breaks. If the **Display non-printable characters** check box is selected, you can press Ctrl/Command+* to enable and disable the display of non-printing characters.

     Spaces and tab characters are displayed in all input fields (except **Description**), in normalizers, correlation rules, filters and connectors, and in SQL queries for searching events in the **Events** section. Spaces are displayed as dots. A tab character is displayed as a dash in normalizers, correlation rules, filters and connectors. In other fields, a tab character is displayed as one or two dots.

     Line break characters are displayed in all input fields that support multi-line input, such as the event search field.
4. 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**.
   - **Current password** (required)—enter the current password of your user account. The field is available if you change your account password.
   - **New password** (required)—enter a new password to the user account. Password requirements:
     - 8 to 128 characters long.
     - At least one lowercase character.
     - At least one uppercase character.
     - At lease one numeral.
     - At least one of the following special characters: !, @, #, %, ^, &, *.
   - **Confirm password** (required)—enter the password again for confirmation.
5. If necessary, use the **Generate token** button to generate an API token. Clicking this button displays the token creation window.
6. If necessary, configure the operations available to the user via the REST API by using the **API access rights** button.
7. Click **Save**.

The user account will be changed.

Editing your user account

To edit your user account:

1. Open the KUMA web interface, click the name of your user account in the bottom-left corner of the window and click the **Profile** button in the opened menu.

   The **User** window with your user account parameters opens.

2. Make the necessary changes to the parameters:

   - **Name** (required)—enter the user name. The length of the string must be 1 to 128 Unicode characters.
   - **Login**(required) – enter a unique user name for the user account. Must contain from 3 to 64 characters (only a–z, A–Z, 0–9, . \ - _).

     Email (required)—enter the unique email address of the user. Must be a valid email address.

3. Select or clear the check boxes that control access rights and user capabilities:

   - **Receive email notifications**—select this check box if you want the user to receive SMTP notifications from KUMA.
   - **Display non-printable characters**—select this check box if you want the KUMA web interface to display non-printing characters such as spaces, tab characters, and line breaks.

     Spaces and tab characters are displayed in all input fields (except **Description**), in normalizers, correlation rules, filters and connectors, and in SQL queries for searching events in the **Events** section.

     Spaces are displayed as dots.

     A tab character is displayed as a dash in normalizers, correlation rules, filters and connectors. In other fields, a tab character is displayed as one or two dots.

     Line break characters are displayed in all input fields that support multi-line input, such as the event search field.

     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.

4. 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**.

   - **Current password** (required)—enter the current password of your user account.
   - **New password** (required)—enter a new password to your account. Password requirements:
     - 8 to 128 characters long.
     - At least one lowercase character.
     - At least one uppercase character.
     - At lease one numeral.
     - At least one of the following special characters: !, @, #, %, ^, &, *.
   - **Confirm password** (required)—enter the password again for confirmation.

5. If necessary, use the **Generate token** button to generate an API token. Clicking this button displays the token creation window.

6. If necessary, configure the operations that are available via the REST API by using the **API access rights** button.

7. Click **Save**.

Your user account is changed.

KUMA services

Services are the main components of KUMA that help the system to manage events: services allow you to receive events from event sources and subsequently bring them to a common form that is convenient for finding correlation, as well as for storage and manual analysis. Each service consists of two parts that work together:

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

  > On the server side, KUMA services are located in the `/opt/kaspersky/kuma` directory.

  When you install KUMA in high availability mode, only the KUMA Core is installed in the cluster. Collectors, correlators, and storages are hosted on hosts outside of the Kubernetes cluster.

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

Service types:

- Storages are used to save events.
- Correlators are used to analyze events and search for defined patterns.
- Collectors are used to receive events and convert them to KUMA format.
- Agents are used to receive events on remote devices and forward them to KUMA collectors.

In the KUMA 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.

The maximum table size is not limited. If you want to select all services, scroll to the end of the table and select the **Select all** check box, which selects all available services in the table.

Table columns:

- **Status**—service status:
  - Green means that the service is running.
  - Red means that the service is not running.
  - Yellow means that there is no connection with ClickHouse nodes (this status is applied only to storage services). The reason for this is indicated in the service log if logging was enabled.
  - Gray—if a deleted tenant had a running service that continues to work, that service is displayed with a gray status on the **Active services** page. Services with the gray status are kept to let you copy the ID and remove services on your servers. Only the General administrator can delete services with the gray status.
- **Type**—type of service: **agent**, **collector**, **correlator**, or **storage**.

- **Name**—name of the service. Clicking on the name of the service opens its settings.
- **Version**—service version.
- **Tenant**—the name of the tenant that owns the service.
- **FQDN**—fully qualified domain name of the service server.
- **IP address**—IP address of the server where the service is installed.
- **API port**—Remote Procedure Call port number.
- **Uptime**—the time showing how long the service has been running.
- **Created**—the date and time when the service was created.

The table can be sorted in ascending and descending order, as well as by the **Status** parameter. To sort active services, right-click the context menu and select one or more statuses.

You can 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.
- Add service to Kubernetes
- Refresh list
- Update configuration
- Restart
- Reset certificate
- Delete

To perform an action with an individual service, right-click the service to display its context menu. The following actions are available:

- Copy service ID

  You need this ID to install, restart, stop, or delete the service.
- Go to Events
- Update service configuration
- Restart service
- Download log

  If you want to receive detailed information, enable the Debug mode in the service settings.
- Reset certificate

- Delete service

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.

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. Select the check box next to the service whose ID you want to obtain, and click **Copy ID**.

   The identifier of the service will be copied to the clipboard. 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 "Commands for stopping, starting, and checking the status of a service" table lists commands that may be useful when managing KUMA.

Commands for stopping, starting, and checking the status of a service

Service	Stop service	Start service	

Core	`sudo systemctl stop kuma-core.service`	`sudo systemctl start kuma-core.service`	suc cor
Services with an ID: - collector - correlator - storage	`sudo systemctl stop kuma-<collector/correlator/storage>-<service ID>.service`	`sudo systemctl start kuma-<collector/correlator/storage>-<service ID>.service`	suc <cc <se
Services without an ID: - kuma-grafana.service - kuma-mongodb.service - kuma-victoria-metrics.service - kuma-vmalert.service	`sudo systemctl stop kuma-<grafana/victoria-metrics/vmalert>.service`	`sudo systemctl start kuma-<grafana/victoria-metrics/vmalert>.service`	suc <gr met
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 that has administrative privileges. 4. Run the following command in PowerShell: `Stop-Service -Name "WindowsAgent-<agent ID>"`	To start an agent service: 1. Copy the agent ID in the KUMA web interface. 2. Connect to the host on which you want to start the KUMA agent service. 3. Run PowerShell as an account that has administrative privileges. 4. Run the following command in PowerShell: `Start-Service -Name "WindowsAgent-<agent ID>"`	To v 1. In Ser ser KUN 2. T viev Ser

Restarting the service

To restart the service:

1. Log in to the KUMA web interface and open **Resources** → **Active services**.
2. Select the check box next to the service and select the necessary option:
   - **Update configuration**—perform a hot update of a running service configuration. For example, you can change the field mapping settings or the destination point settings this way.
   - **Restart**—stop a service and start it again. This option is used to modify the port number or connector type.

     Restarting KUMA agents:

  - KUMA Windows Agent can be restarted as described above only if it is running on a remote computer. If the service on the remote computer is inactive, you will receive an error when trying to restart from KUMA. In that case you must restart KUMA Windows Agent service on the remote Windows machine. For information on restarting Windows services, refer to the documentation specific to the operating system version of your remote Windows computer.
  - KUMA Agent for Linux stops when this option is used. To start the agent again, you must execute the command that was used to start it.
- **Reset certificate**—remove certificates that the service uses for internal communication. For example, this option can be used to renew the Core certificate.

  Special considerations for deleting Windows agent certificates:

  - If the agent has the green status and you select **Reset certificate**, KUMA deletes the current certificate and creates a new one, the agent continues working with the new certificate.
  - If the agent has the red status and you select **Reset certificate**, KUMA generates an error that the agent is not running. In the agent installation folder %APPDATA%\kaspersky\kuma\<Agent ID>\certificates, manually delete the internal.cert and internal.key files and start the agent manually. When the agent starts, a new certificate is created automatically.

  Special considerations for deleting Linux agent certificates:

  1. Regardless of the agent status, apply the **Reset certificate** option in the web interface to delete the certificate in the databases.
  2. In the agent installation folder **`/opt/kaspersky/agent/<Agent ID>/certificates`**, manually delete the internal.cert and internal.key files.
  3. Since the **Reset certificate** option stops the agent, to continue its operation, start the agent manually. When the agent starts, a new certificate is created automatically.

Deleting the service

Before deleting the service get its ID. The ID will be required to remove the service for the server.

To remove a service in the KUMA web interface:

1. Log in to the KUMA web interface and open **Resources** → **Active services**.
2. Select the check box next to the service you want to delete, and click **Delete**.

   A confirmation window opens.
3. Click **OK**.

The service has been deleted from KUMA.

To remove a service from the server, run the following command:

```
sudo /opt/kaspersky/kuma/kuma <collector/correlator/storage> --id < service ID > --
uninstall
```

```
The service has been deleted from the server.
```

Partitions window

If the storage service was created and installed, you can view its partitions in the **Partitions** table.

*To open **Partitions** table:*

1. Log in to the KUMA web interface and open **Resources** → **Active services**.
2. Select the check box next to the relevant storage and click **Go to partitions**.

   The **Partitions** table opens.

The table has the following columns:

- **Tenant**—the name of the tenant that owns the stored data.
- **Created**—partition creation date.
- **Space**—the name of the space.
- **Size**—the size of the space.
- **Events**—the number of stored events.
- **Transfer to cold storage**—the date when data will be migrated from the ClickHouse clusters to cold storage disks.
- **Expires**—the date when the partition expires. After this date, the partition and the events it contains are no longer available.

You can delete partitions.

To delete a partition:

1. Open the **Partitions** table (see above).
2. Open the ... drop-down list to the left from the required partition.
3. Select **Delete**.
   A confirmation window opens.
4. Click **OK**.

   The partition has been deleted. Audit event partitions cannot be deleted.

Searching for related events

You can search for events processed by the Correlator or the Collector services.

To search for events related to the Correlator or the Collector service:

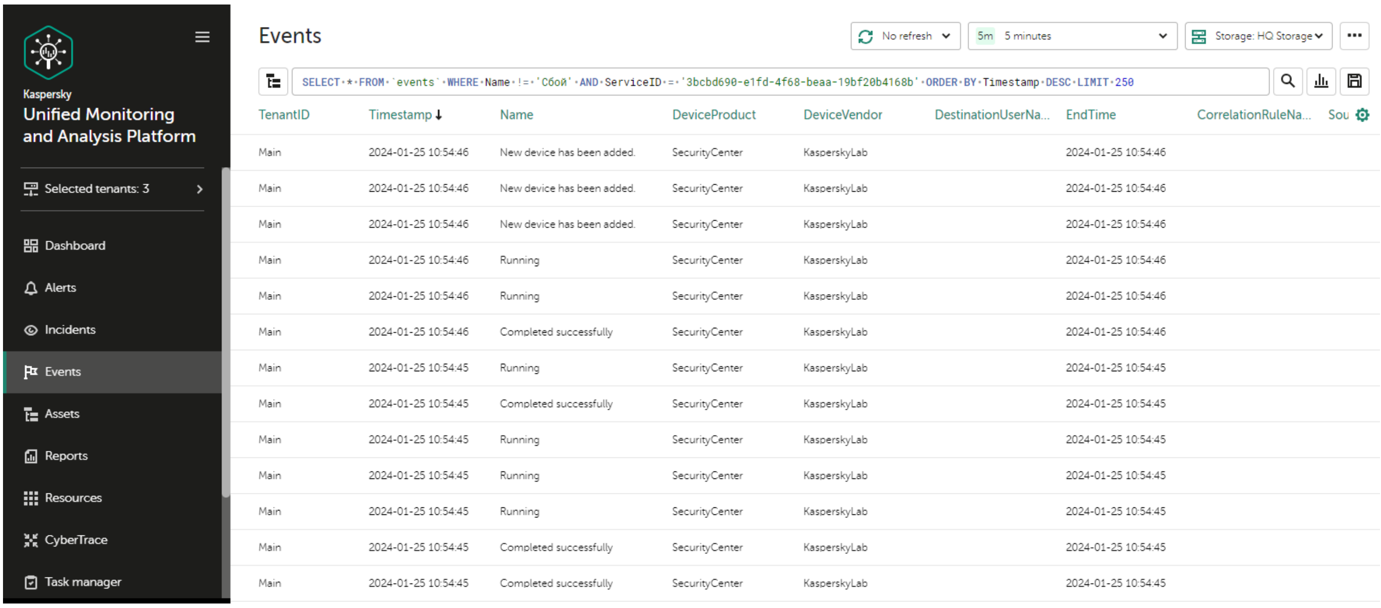
1. Log in to the KUMA web interface and open **Resources** → **Active services**.

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.



Events

```
SELECT * FROM `events` WHERE Name != 'Сбой' AND ServiceID = '3bcbd690-e1fd-4f68-beaa-19bf20b4168b' ORDER BY Timestamp DESC LIMIT 250
```

TenantID	Timestamp ↓	Name	DeviceProduct	DeviceVendor	DestinationUserNa...	EndTime	CorrelationRuleNa...	Sou
Main	2024-01-25 10:54:46	New device has been added.	SecurityCenter	KasperskyLab		2024-01-25 10:54:46		
Main	2024-01-25 10:54:46	New device has been added.	SecurityCenter	KasperskyLab		2024-01-25 10:54:46		
Main	2024-01-25 10:54:46	New device has been added.	SecurityCenter	KasperskyLab		2024-01-25 10:54:46		
Main	2024-01-25 10:54:46	Running	SecurityCenter	KasperskyLab		2024-01-25 10:54:46		
Main	2024-01-25 10:54:46	Running	SecurityCenter	KasperskyLab		2024-01-25 10:54:46		
Main	2024-01-25 10:54:46	Completed successfully	SecurityCenter	KasperskyLab		2024-01-25 10:54:46		
Main	2024-01-25 10:54:45	Running	SecurityCenter	KasperskyLab		2024-01-25 10:54:45		
Main	2024-01-25 10:54:45	Completed successfully	SecurityCenter	KasperskyLab		2024-01-25 10:54:45		
Main	2024-01-25 10:54:45	Running	SecurityCenter	KasperskyLab		2024-01-25 10:54:45		
Main	2024-01-25 10:54:45	Running	SecurityCenter	KasperskyLab		2024-01-25 10:54:45		
Main	2024-01-25 10:54:45	Running	SecurityCenter	KasperskyLab		2024-01-25 10:54:45		
Main	2024-01-25 10:54:45	Completed successfully	SecurityCenter	KasperskyLab		2024-01-25 10:54:45		
Main	2024-01-25 10:54:45	Completed successfully	SecurityCenter	KasperskyLab		2024-01-25 10:54:45		

Event search results

Service resource sets

Service resource sets are a resource type, a KUMA component, a set of settings based on which the KUMA services are created and operate. Resource sets for services are collections of resources.

Any resources added to a set of resources must be owned by the same tenant that owns the created set of resources. An exception is the shared tenant, whose owned resources can be used in the sets of resources of other tenants.

Resource sets for services are displayed in the **Resources** → **<Resource set type for the service>** section of the KUMA 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.

It is recommended to use ext4 as the file system.

A storage is created in several steps:

1. Creating a set of resources for a storage in the KUMA 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.

If the storage settings are changed, the service must be restarted.

ClickHouse cluster structure

A ClickHouse *cluster* is a logical group of devices that possess all accumulated normalized KUMA events. It consists of one or more logical *shards*.

A *shard* is a logical group of devices that possess a specific **portion** of all normalized events accumulated in the cluster. It consists of one or more *replicas*. Increasing the number of shards lets you do the following:

- Accumulate more events by increasing the total number of servers and disk space.
- Absorb a larger **stream** of events by distributing the load associated with an influx of new events.
- Reduce the time taken to search for events by distributing search zones among multiple devices.

A *replica* is a device that is a member of a logical shard and possesses a single copy of that shard's data. If multiple replicas exist, it means multiple copies exist (the data is replicated). Increasing the number of replicas lets you do the following:

- Improve high availability.
- Distribute the total load related to data searches among multiple machines (although it'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 or the Hadoop Distributed File System (HDFS). Cold storage is enabled when at least one cold storage disk is specified. If a cold storage disk is not configured and the server runs out of disk space, the storage service is stopped. If both hot storage and cold storage are configured, and space runs out on the cold storage disk, the KUMA storage service is stopped. We recommend avoiding such situations.

Cold storage disks can be added or removed.

After changing the cold storage settings, the storage service must be restarted. If the service does not start, the reason is specified in the storage log.

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 disks) and then delete it from the storage settings.

Rules for moving the data to the cold storage disks

When cold storage is enabled, KUMA checks the storage terms of the spaces once an hour:

- If the storage term for a space on a ClickHouse cluster expires, the data is moved to the cold storage disks. If a cold storage disk is misconfigured, the data is deleted.
- If the storage term for a space on a cold storage disk expires, the data is deleted.
- If the ClickHouse cluster disks are 95% full, the biggest partitions are automatically moved to the cold storage disks. This can happen more often than once per hour.
- Audit events are generated when data transfer starts and ends.

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 the mouse pointer over the status icon, a message indicating the data transfer appears. 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.
- Storing events or audit events on cold storage disks is not mandatory; to disable this functionality, specify `0` (days) in the **Cold retention period** or **Audit cold retention period** field in the storage settings.

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.

Removing cold storage disks

Before physically disconnecting cold storage disks, remove these disks from the storage settings.

To remove a disk from the storage settings:

- In the KUMA web interface, under **Resources** → **Storages**, select the relevant storage.

  This opens the storage.

- In the window, in the **Disks for cold storage** section, in the required disk's group of settings, click **Delete disk**.

  Data from removed disk is automatically migrated to other cold storage disks or, if there are no such disks, to the ClickHouse cluster. During data migration, the storage status icon is highlighted in yellow. Audit events are generated when data transfer starts and ends.

- After event migration is complete, the disk is automatically removed from the storage settings. It can now be safely disconnected.

Removed disks can still contain events. If you want to delete them, you can manually delete the data partitions using the DROP PARTITION command.

If the cold storage disk specified in the storage settings has become unavailable (for example, out of order), this may lead to errors in the operation of the storage service. In this case, create a disk with the same path (for local disks) or the same address (for HDFS disks) and then delete it from the storage settings.

Detaching, archiving, and attaching partitions

If you want to optimize disk space and speed up queries in KUMA, you can detach data partitions in ClickHouse, archive partitions, or move partitions to a drive. If necessary, you can later reattach the partitions you need and perform data processing.

Detaching partitions

To detach partitions:

1. Determine the shard on all replicas of which you want to detach the partition.

2. Get the partition ID using the following command:

   ```
   sudo /opt/kaspersky/kuma/clickhouse/bin/client.sh -d kuma --multiline --query "SELECT partition, name FROM system.parts;" |grep 20231130
   ```

   In this example, the command returns the partition ID for November 30, 2023.

3. One each replica of the shard, detach the partition using the following command and specifying the partition ID:

   ```
   sudo /opt/kaspersky/kuma/clickhouse/bin/client.sh -d kuma --multiline --query "ALTER TABLE events_local_v2 DETACH PARTITION ID '<partition ID>'"
   ```

As a result, the partition is detached on all replicas of the shard. Now you can move the data directory to a drive or archive the partition.

Archiving partitions

To archive detached partitions:

1. Find the detached partition in disk subsystem of the server:

   ```
   sudo find /opt/kaspersky/kuma/clickhouse/data/ -name <ID of the detached partition>\*
   ```

2. Change to the 'detached' directory that contains the detached partition, and while in the 'detached' directory, perform the archival:

   ```
   sudo cd <path to the 'detached' directory containing the detached partition>
   ```

   ```
   sudo zip -9 -r detached.zip *
   ```

   For example:

   ```
   sudo cd /opt/kaspersky/kuma/clickhouse/data/store/d5b/d5bdd8d8-e1eb-4968-95bd-d8d8e1eb3968/detached/
   ```

   ```
   sudo zip -9 -r detached.zip *
   ```

The partition is archived.

Attaching partitions

To attach archived partitions to KUMA:

1. Increase the **Retention period** value.

   KUMA deletes data based on the date specified in the Timestamp field, which records the time when the event is received, and based on the **Retention period** value that you set for the storage.

   Before restoring archived data, make sure that the **Retention period** value overlaps the date in the Timestamp field. If this is not the case, the archived data will be deleted within 1 hour.

2. Place the archive partition in the 'detached' section of your storage and unpack the archive:

   `sudo` unzip detached.zip -d `<path to the 'detached' directory>`

   For example:

   `sudo` unzip detached.zip -d `/opt/kaspersky/kuma/clickhouse/data/store/d5b/d5bdd8d8-e1eb-4968-95bd-d8d8e1eb3968/detached/`

3. Run the command to attach the partition:

   ```
   sudo /opt/kaspersky/kuma/clickhouse/bin/client.sh -d kuma --multiline --query "ALTER TABLE events_local_v2 ATTACH PARTITION ID '<partition ID>'"
   ```

   `Repeat the steps of unpacking the archive and attaching the partition on each replica of the shard.`

As a result, the archived partition is attached and its events are again available for search.

Creating a set of resources for a storage

In the KUMA web interface, a storage service is created based on the set of resources for the storage.

To create a set of resources for a storage in the KUMA web interface:

1. In the KUMA web interface, under **Resources** → **Storages**, click **Add storage**.

   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. You can optionally add up to 256 Unicode characters describing the service in the **Description** field.

5. In the **Retention period** field, specify the period, in days from the moment of arrival, during which you want to store events in the ClickHouse cluster. When the specified period expires, events are automatically deleted from the ClickHouse cluster. If cold storage of events is configured, when the event storage period in the ClickHouse cluster expires, the data is moved to cold storage disks. If a cold storage disk is misconfigured, the data is deleted.

6. In the **Audit retention period** field, specify the period, in days, to store audit events. The minimum value and default value is `365`.

7. If cold storage is required, specify the event storage term:

   - **Cold retention period**—the number of days to store events. The minimum value is `1`.

   - **Audit cold retention period**—the number of days to store audit events. The minimum value is 0.

8. Use the **Debug** toggle switch to specify whether resource logging must be enabled. The default value (**Disabled**) means that only errors are logged for all KUMA components. If you want to obtain detailed data in the logs, select **Enabled**.

9. 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>

10. 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 and remove them by clicking the **Delete space** button.

    Available settings:

    - In the **Name** field, specify a name for the space containing 1 to 128 Unicode characters.

    - In the **Retention period** field, specify the number of days to store events in the ClickHouse cluster.

    - If necessary, in the **Cold retention period** field, specify the number of days to store the events in the cold storage. The minimum value is `1`.

    - In the **Filter** section, you can specify conditions to identify events that will be put into this space. You can select an existing filter from the drop-down list or **create** a new filter.

      Creating a filter in resources ?

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. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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.

11. 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 **Delete disk** button.

Available settings:

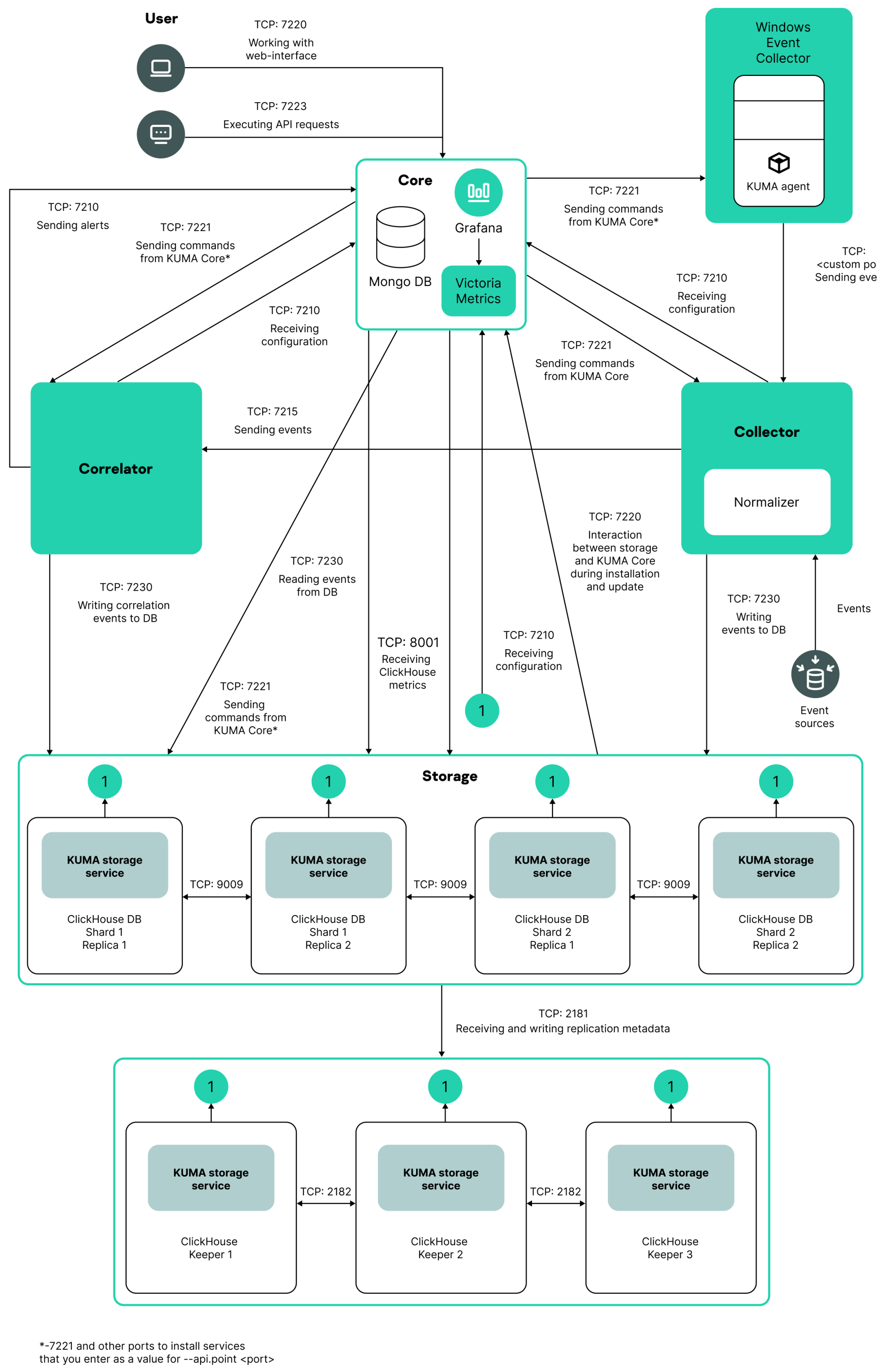
- In the **Type** drop-down list, select the type of the disk being connected:
  - **Local**—for the disks mounted in the operating system as directories.
  - **HDFS**—for the disks of the Hadoop Distributed File System.
- In the **Name** field, specify the disk name. The name must contain 1 to 128 Unicode characters.
- If you select **Local** disk type, specify the absolute directory path of the mounted local disk in the **Path** field. The path must begin and end with a "/" character.
- If you select **HDFS** disk type, specify the path to HDFS in the **Host** field. For example, `hdfs://hdfs1:9000/clickhouse/`.

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 and remove them by clicking the **Remove node** button.

Available settings:

- In the **FQDN** field, specify the fully qualified domain name of the node being added. For example, `kuma-storage-cluster1-server1.example.com`.
- In the shard, replica, 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 the example to suit 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. In version 2.1.3 or later, the **Advanced Settings** tab is available. On the **Advanced settings** tab, in the **Buffer size** field, enter the buffer size in bytes, that causes events to be sent to the database when reached. The default value is 64 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.

14. On the **Advanced Settings** tab, 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.

15. On the **Advanced settings** tab, in the **Disk buffer size limit** field, enter the 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.

16. On the **Advanced Settings** tab, from the **Disk buffer disabled** drop-down list, select a value to **Enable** or **Disable** the use of the disk buffer. By default, the disk buffer is enabled.

17. On the **Advanced Settings** tab, In the **Write to local database table** drop-down list, select **Enable** or **Disable**. Writing is disabled by default.

    In **Enable** mode, data is written only on the host where 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**.

    In **Disable** mode, data is distributed among the shards of the cluster.

The set of resources for the storage is created and is displayed under **Resources** → **Storages**. Now you can create a storage service.

Creating a storage service in the KUMA web interface

When a set of resources is created for a storage, you can proceed to create a storage service in KUMA.

To create a storage service in the KUMA 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 set of resources 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

To create a storage:

1. Log in to the server where you want to install the service.
2. Create the /opt/kaspersky/kuma/ folder.
3. Copy the "kuma" file to the /opt/kaspersky/kuma/ folder. The file is located in the installer in the /kuma-ansible-installer/roles/kuma/files/ folder.

   > Make sure the kuma file has sufficient rights to run.

4. Execute the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma storage --core https://<KUMA Core server FQDN>:<port used by KUMA Core for internal communication (port 7210 by default)> --id <service ID copied from the KUMA 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`.

5. Repeat steps 1–2 for each storage node.

   The storage is installed.

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 set of resources for the correlator. Upon completion of the Wizard, the service is automatically created based on this set of resources.

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 define the main settings of the correlator:

- In the **Name** field, enter a unique name for the service you are creating. The name must contain 1 to 128 Unicode characters.
- In the **Tenant** drop-down list, select the tenant that will own the correlator. The tenant selection determines what resources will be available when the collector is created.

  > If you return to this window from another subsequent step of the Installation Wizard and select another tenant, you will have to manually edit all the resources that you have added to the service. Only resources from the selected tenant and shared tenant can be added to the service.

- If required, specify the number of processes that the service can run concurrently in the **Workers** field. By default, the number of worker processes is the same as the number of vCPUs on the server where the service is installed.
- If necessary, use the **Debug** toggle switch to enable logging of service operations.
- You can optionally add up to 256 Unicode characters describing the service in the **Description** field.

The main settings of the correlator are defined. Proceed to the next step of the Installation Wizard.

Step 2. Global variables

If tracking values in event fields, active lists, or dictionaries is not enough to cover some specific security scenarios, you can use global and local variables. You can use them to take various actions on the values received by the correlators by implementing complex logic for threat detection. Variables can be assigned a specific function and then queried from correlation rules as if they were ordinary event fields, with the triggered function result received in response.

To add a global variable in the correlator,

click the **Add variable** button and specify the following parameters:

- In the **Variable** window, enter the name of the variable.

  Variable naming requirements ?

- Must be unique within the correlator.
- Must contain 1 to 128 Unicode characters.
- Must not begin with the character $.
- Must be written in camelCase or CamelCase.

- In the **Value** window, enter the variable function.

  Description of variable functions.

The global variable is added. It can be queried from correlation rules by adding the $ character in front of the variable name. There can be multiple variables. Added variables can be edited or deleted by using the × icon.

Proceed to the next step of the Installation Wizard.

Step 3. Correlation

This is an optional but recommended step of the Installation Wizard. On the **Correlation** tab of the Installation Wizard, select or create correlation rules. These resources define the sequences of events that indicate security-related incidents. When these sequences are detected, the correlator creates a correlation event and an alert.

If you have added global variables to the correlator, all added correlation rules can query them.

Correlation rules that are added to the set of resources for the correlator are displayed in the table with the following columns:

- **Correlation rules**—name of the correlation rule resource.
- **Type**—type of correlation rule: **standard**, **simple**, **operational**. The table can be filtered based on the values of this column by clicking the column header and selecting the relevant values.
- **Actions**—list of actions that will be performed by the correlator when the correlation rule is triggered. These actions are indicated in the correlation rule settings. The table can be filtered based on the values of this column by clicking the column header and selecting the relevant values.

  Available values:

  - **Output**—correlation events created by this correlation rule are transmitted to other correlator resources: enrichment, response rule, and then to other KUMA services.
  - **Edit active list**—the correlation rule changes the active lists.
  - **Loop to correlator**—the correlation event is sent to the same correlation rule for reprocessing.
  - **Categorization**—the correlation rule changes asset categories.
  - **Event enrichment**—the correlation rule is configured to enrich correlation events.
  - **Do not create alert**—when a correlation event is created as a result of a correlation rule triggering, no alert is created for that.
  - **Shared resource**—the correlation rule or the resources used in the correlation rule are located in a shared tenant.

You can use the **Search** field to search for a correlation rule. Added correlation rules can be removed from the set of resources by selecting the relevant rules and clicking **Delete**.

Selecting a correlation rule opens a window with its settings, which can be edited and then saved by clicking **Save**. If you click **Delete** in this window, the correlation rule is unlinked from the set of resources.

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 set of resources for the correlator:

1. Click **Link**.

   The resource selection window opens.

2. Select the relevant correlation rules and click **OK**.

The correlation rules will be linked to the set of resources for the correlator and will be displayed in the rules table.

To create a new correlation rule in a set of resources for a correlator:

1. Click **Add**.

   The correlation rule creation window opens.

2. Specify the correlation rule settings and click **Save**.

The correlation rule will be created and linked to the set of resources for the correlator. It is displayed in the correlation rules table and in the list of resources under **Resources** → **Correlation rules**.

Proceed to the next step of the Installation Wizard.

Step 4. Enrichment

This is an optional step of the Installation Wizard. On the **Enrichment** tab of the Installation Wizard, you can select or create a enrichment rules and indicate which data from which sources you want to add to correlation events that the correlator creates. There can be more than one enrichment rule. You can add them by clicking the **Add** button and can remove them by clicking the × button.

To add an existing enrichment rule to a set of resources:

1. Click **Add**.

   This opens the enrichment rule settings block.

2. In the **Enrichment rule** drop-down list, select the relevant resource.

The enrichment rule is added to the set of resources for the correlator.

To create a new enrichment rule in a set of resources:

1. Click **Add**.

   This opens the enrichment rule settings block.

2. In the **Enrichment rule** drop-down list, select **Create new**.

3. In the **Source kind** drop-down list, select the source of data for enrichment and define its corresponding settings:

   - constant

     This type of enrichment is used when a constant needs to be added to an event field. Settings of this type of enrichment:

     - In the **Constant** field, specify the value that should be added to the event field. The value may not be longer than 255 Unicode characters. If you leave this field blank, the existing event field value will be cleared.

     - In the **Target field** drop-down list, select the KUMA event field to which you want to write 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.

     When this type is selected in the **Dictionary name** drop-down list, you must select the dictionary that will provide the values. In the **Key fields** settings block, you must use the **Add field** button to select the event fields whose values will be used for dictionary entry selection.

     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 3 elements: "a", "b" and "c". The following value is 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 two fields: the SA.StringArrayOne extended schema field and the Code field. The SA.StringArrayOne extended schema field contains 3 elements: "a", "b", and "c"; the Code string field contains the character sequence "myCode". The following value is passed to the dictionary as the key: ['a','b','c']|myCode.

   - event

This type of enrichment is used when you need to write a value from another event field to the current event field. Settings of this type of enrichment:

- In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.
- In the **Source field** drop-down list, select the event field whose value will be written to the target field.
- In the **Conversion** settings block, you can create rules for modifying the original data before it is written to the KUMA event fields. The conversion type can be selected from the drop-down list. You can use the **Add conversion** and **Delete** buttons to add or delete a conversion, respectively. The order of conversions is important.

  Available conversions

Conversions are changes that can be applied to a value before it gets written to the event field. The conversion type is selected from a drop-down list.

Available conversions:

- **lower**—is used to make all characters of the value lowercase
- **upper**—is used to make all characters of the value uppercase
- **regexp** – used to convert a value using the regular expression RE2. When this conversion type is selected, the field appears where regular expression should be added.
- **substring**—is used to extract characters in the position range specified in the **Start** and **End** fields. These fields appear when this conversion type is selected.
- **replace**—is used to replace specified character sequence with the other character sequence. When this type of conversion is selected, new fields appear:
  - **Replace chars**—in this field you can specify the character sequence that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- **trim**—used to simultaneously remove the characters specified in the **Chars** field from the leading and end positions of the value. The field appears when this type of conversion is selected. For example, a **trim** conversion with the `Micromon` value applied to `Microsoft-Windows-Sysmon` results in `soft-Windows-Sys`.
- **append** is used to add the characters specified in the **Constant** field to the end of the event field value. The field appears when this type of conversion is selected.
- **prepend**—used to prepend the characters specified in the **Constant** field to the start of the event field value. The field appears when this type of conversion is selected.
- **replace with regexp**—is used to replace RE2 regular expression results with the character sequence.
  - **Expression**—in this field you can specify the regular expression which results that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, such a string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the "String" type, all types of conversions are available.
- For fields of the "Number" and "Float" types, the following types of conversions are available: regexp, substring, replace, trim, append, prepend, replaceWithRegexp, decodeHexString, decodeBase64String, decodeBase64URLString.
- For fields of "Array of strings", "Array of numbers", and "Array of floats" types, the following types of conversions are available: append, prepend.

- template

  This type of enrichment is used when you need to write a value obtained by processing Go templates into the event field. Settings of this type of enrichment:

  - Put the Go template into the **Template** field.

    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.

    Example: `Attack on {{.DestinationAddress}} from {{.SourceAddress}}`.
  - In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.

  If you are using enrichment of events that have the "Template" type selected as the "Source kind" setting, 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.

  Example:

  {{.SA.StringArrayOne}}

  Example:

  {{- range $index, $element := . SA.StringArrayOne -}}

  {{- if $index}}, {{end}}"{{$element}}"{{- end -}}

- dns

This type of enrichment is used to send requests to a private network DNS server to convert IP addresses into domain names or vice versa. IP addresses are converted to DNS names only for private addresses: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 100.64.0.0/10.

Available settings:

- **URL**—in this field, you can specify the URL of a DNS server to which you want to send requests. You can use the **Add URL** button to specify multiple URLs.
- **RPS**—maximum number of requests sent to the server per second. The default value is `1,000`.
- **Workers**—maximum number of requests per one point in time. The default value is `1`.
- **Max tasks**—maximum number of simultaneously fulfilled requests. By default, this value is equal to the number of vCPUs of the KUMA Core server.
- **Cache TTL**—the lifetime of the values stored in the cache. The default value is `60`.
- **Cache disabled**—you can use this drop-down list to enable or disable caching. Caching is enabled by default.

- cybertrace

  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.
  - **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`.
  - **Mapping** (required)—this settings block contains the mapping table for mapping KUMA event fields to CyberTrace indicator types. The **KUMA field** column shows the names of KUMA event fields, and the **CyberTrace indicator** column shows the types of CyberTrace indicators.

    Available types of CyberTrace indicators:

    - ip
    - url
    - hash

    In the mapping table, you must provide at least one string. You can use the **Add row** button to add a string, and can use the ✕ button to remove a string.

- timezone

This type of enrichment is used in collectors and correlators to assign a specific timezone to an event. Timezone information may be useful when searching for events that occurred at unusual times, such as nighttime.

When this type of enrichment is selected, the required timezone must be selected from the **Timezone** drop-down list.

> Make sure that the required time zone is set on the server hosting the enrichment-utilizing service. For example, you can do this by using the `timedatectl list-timezones` command, which shows all time zones that are set on the server. For more details on setting time zones, please refer to your operating system documentation.

When an event is enriched, the time offset of the selected timezone relative to Coordinated Universal Time (UTC) is written to the DeviceTimeZone event field in the `+-hh:mm` format. For example, if you select the **Asia/Yekaterinburg** timezone, the value `+05:00` will be written to the DeviceTimeZone field. If the enriched event already has a value in the DeviceTimeZone field, it will be overwritten.

By default, if the timezone is not specified in the event being processed and enrichment rules by timezone are not configured, the event is assigned the timezone of the server hosting the service (collector or correlator) that processes the event. If the server time is changed, the service must be restarted.

Permissible time formats when enriching the DeviceTimeZone field ?

When processing incoming raw events in the collector, the following time formats can be automatically converted to the +-hh:mm format:

Time format in a processed event	Example
+-hh:mm	-07:00
+-hhmm	-0700
+-hh	-07

If the date format in the `DeviceTimeZone` field differs from the formats listed above, the collector server timezone is written to the field when an event is enriched with timezone information. You can create custom normalization rules for non-standard time formats.

4. Use the **Debug** 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 ?

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. The name must contain 1 to 128 Unicode characters.
4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

   Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

   Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search condition (and, or, not) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button.

   You can view the nested filter settings by clicking the ↗ button.

The new enrichment rule was added to the set of resources for the correlator.

Proceed to the next step of the Installation Wizard.

Step 5. Response

This is an optional step of the Installation Wizard. On the **Response** tab of the Installation Wizard, you can select or create response rules and indicate which actions must be performed when the correlation rules are triggered. There can be multiple response rules. You can add them by clicking the **Add** button and can remove them by clicking the ✕ button.

To add an existing response rule to a set of resources:

1. Click **Add**.

   The response rule settings window opens.

2. In the **Response rule** drop-down list, select the relevant resource.

The response rule is added to the set of resources for the correlator.

To create a new response rule in a set of resources:

1. Click **Add**.

   The response rule settings window opens.

2. In the **Response rule** drop-down list, select **Create new**.

3. In the **Type** drop-down list, select the type of response rule and define its corresponding settings:

   - **KSC response**—response rules for automatically launching the tasks on Kaspersky Security Center assets. For example, you can configure automatic startup of a virus scan or database update.

Tasks are automatically started when KUMA is integrated with Kaspersky Security Center. Tasks are run only on assets that were imported from Kaspersky Security Center.

Response settings

- **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 folder, the service will not start.

  - **Script arguments**—parameters or event field values that must be passed to the script.

    > If the script includes actions taken on files, you should specify the absolute path to these files.

    Parameters can be written with quotation marks (").

    Event field names are passed in the `{{.EventField}}` format, where `EventField` is the name of the event field which value must be passed to the script.

    Example: `-n "\"usr\": {{.SourceUserName}}"`

- **KEDR response**—response rules for automatically creating prevention rules, starting network isolation, or starting the application on Kaspersky Endpoint Detection and Response and Kaspersky Security Center assets.

  Automatic response actions are carried out when KUMA is integrated with Kaspersky Endpoint Detection and Response.

  Response settings ?

- **Event field** (required)—event field containing the asset for which the response actions are needed. Possible values:
  - SourceAssetID
  - DestinationAssetID
  - DeviceAssetID
- **Task type**—response action to be performed when data matching the filter is received. The following types of response actions are available:
  - Enable network isolation.

    When selecting this type of response, you need to define values for the following settings:

    - **Isolation timeout**—the number of hours during which the network isolation of an asset will be active. You can indicate from 1 to 9,999 hours.

      If necessary, you can add an exclusion for network isolation ?.

      To add an exclusion for network isolation:

      a. Click the **Add exclusion** button.

      b. Select the direction of network traffic that must not be blocked:

         - Inbound.
         - Outbound.
         - Inbound/Outbound.

      c. In the **Asset IP** field, enter the IP address of the asset whose network traffic must not be blocked.

      d. If you selected **Inbound** or **Outbound**, specify the connection ports in the **Remote ports** and **Local ports** fields.

      e. If you want to add more than one exclusion, click **Add exclusion** and repeat the steps to fill in the **Traffic direction**, **Asset IP**, **Remote ports** and **Local ports** fields.

      f. If you want to delete an exclusion, click the **Delete** button under the relevant exclusion.

      When adding exclusions to a network isolation rule, Kaspersky Endpoint Detection and Response may incorrectly display the port values in the rule details. This does not affect application performance. For more details on viewing a network isolation rule, please refer to the *Kaspersky Anti Targeted Attack Platform Help Guide*.
  - Disable network isolation.
  - Add prevention rule.

    When selecting this type of response, you need to define values for the following settings:

    - **Event fields to extract hash from**—event fields from which 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 program web interface. **Scheduled task** is displayed for this task in the **Created** column of the task table. You can view task completion results.

    > All of the listed operations can be performed on assets that have Kaspersky Endpoint Agent for Windows. On assets that have Kaspersky Endpoint Agent for Linux, the program can only be started.

    > At the software level, the capability to create prevention rules and network isolation rules for assets with Kaspersky Endpoint Agent for Linux is unlimited. KUMA and Kaspersky Endpoint Detection and Response do not provide any notifications about unsuccessful application of these rules.

- **Response via KICS for Networks**—response rules for automatically starting tasks on KICS for Networks assets. For example, you can change the asset status in KICS for Networks.

  Tasks are automatically started when KUMA is integrated with KICS for Networks.

  Response settings ?

- **Event field** (required)—event field containing the asset for which the response actions are needed. Possible values:
  - SourceAssetID
  - DestinationAssetID
  - DeviceAssetID
- **KICS for Networks 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 for Networks 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

- In the **Workers** field, specify the number of processes that the service can run simultaneously.

  By default, the number of workers is the same as the number of virtual processors on the server where the service is installed.

  This field is optional.

1. In the **Filter** section, you can specify conditions to identify events that will be processed using the response rule. You can select an existing filter from the drop-down list or **create** a new filter.

   Creating a filter in resources ⍰

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. The name must contain 1 to 128 Unicode characters.
4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search condition (and, or, not) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button.

   You can view the nested filter settings by clicking the ↗ button.

The new response rule was added to the set of resources for the correlator.

Proceed to the next step of the Installation Wizard.

Step 6. Routing

This is an optional step of the Installation Wizard. On the **Routing** tab of the Installation Wizard, you can select or create destinations with settings indicating the forwarding destination of events created by the correlator. Events from a correlator are usually redirected to storage so that they can be saved and later viewed if necessary. Events can be sent to other locations as needed. There can be more than one destination point.

To add an existing destination to a set of resources for a correlator:

1. In the **Add destination** drop-down list, select the type of destination resource you want to add:

   - Select **Storage** if you want to configure forwarding of processed events to the storage.
   - Select **Correlator** if you want to configure forwarding of processed events to a correlator.
   - Select **Other** if you want to send events to other locations.

     This type of resource includes correlator and storage services that were created in previous versions of the program.

   The **Add destination** window opens where you can specify parameters for events forwarding.

2. In the **Destination** drop-down list, select the necessary destination.

   The window name changes to **Edit destination**, and it displays the settings of the selected resource. The resource can be opened for editing in a new browser tab 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 set of resources for a correlator:

1. In the **Add destination** drop-down list, select the type of destination resource you want to add:
   - Select **Storage** if you want to configure forwarding of processed events to the storage.
   - Select **Correlator** if you want to configure forwarding of processed events to a correlator.
   - Select **Other** if you want to send events to other locations.

     > This type of resource includes correlator and storage services that were created in previous versions of the program.

   The **Add destination** window opens where you can specify parameters for events forwarding.

2. Specify the settings on the **Basic settings** tab:
   - In the **Destination** drop-down list, select **Create new**.
   - In the **Name** field, enter a unique name for the destination resource. The name must contain 1 to 128 Unicode characters.
   - Use the **Disabled** toggle button to specify whether events will be sent to this destination. By default, sending events is enabled.
   - Select the **Type** for the destination resource:
     - Select **storage** if you want to configure forwarding of processed events to the storage.
     - Select **correlator** if you want to configure forwarding of processed events to a correlator.
     - Select **nats-jetstream**, **tcp**, **http**, **kafka**, or **file** if you want to configure sending events to other locations.
   - Specify the **URL** to which events should be sent in the hostname:<API port> format.

     You can specify multiple destination addresses 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. The default value is 1 MB, and the maximum value is 64 MB.

- **Timeout** field is used to set the timeout (in seconds) for another service or component response. The default value is `30`.
- **Disk buffer size limit** field is used to specify the size of the disk buffer in bytes. The default size is 10 GB.
- **Cluster ID** is the ID of the NATS cluster.
- **TLS mode** is a drop-down list where you can specify the conditions for using TLS encryption:
  - **Disabled** (default)—do not use TLS encryption.
  - **Enabled**—encryption is enabled, but without verification.
  - **With verification**—use encryption with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during program installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/.

  When using TLS, it is impossible to specify an IP address as a URL.
- **URL selection policy** is a drop-down list in which you can select a method for determining which URL to send events to if several URLs have been specified:
  - **Any**. Events are sent to one of the available URLs as long as this URL receives events. If the connection is broken (for example, the receiving node is disconnected) a different URL will be selected as the events destination.
  - **Prefer first**. Events are sent to the first URL in the list of added addresses. If it becomes unavailable, events are sent to the next available node in sequence. When the first URL becomes available again, events start to be sent to it again.
  - **Balanced** means that packages with events are evenly distributed among the available URLs from the list. Because packets are sent either on a destination buffer overflow or on the flush timer, this URL selection policy does not guarantee an equal distribution of events to destinations.
- **Delimiter** is used to specify the character delimiting the events. By default, `\n` is used.
- **Path**—the file path if the **file** destination type is selected.
- **Buffer flush interval**—this field is used to set the time interval (in seconds) at which the data is sent to the destination. The default value is `100`.
- **Workers**—this field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
- You can set health checks using the **Health check path** and **Health check timeout** fields. You can also disable health checks by selecting the **Health Check Disabled** check box.
- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.
- The **Disk buffer disabled** drop-down list is used to enable or disable the use of a disk buffer. By default, the disk buffer is disabled.
- In the **Filter** section, you can specify the conditions to define events that will be processed by this resource. You can select an existing filter from the drop-down list or **create** a new filter.

  Creating a filter in resources ?

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. The name must contain 1 to 128 Unicode characters.
4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

   Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

   Filter operators

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.

        - **intersect**—presence in the left operand of the list items specified in the right operand.

    d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

    The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

    This check box is cleared by default.

    e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

    f. You can add multiple conditions or a group of conditions.

  5. If you have added multiple conditions or groups of conditions, choose a search condition (and, or, not) by clicking the **AND** button.

  6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button.

  You can view the nested filter settings by clicking the ↗ button.

4. Click **Save**.

The created destination is displayed on the Installation Wizard tab. A destination resource can be removed from the resource set by selecting it and clicking **Delete** in the opened window.

Proceed to the next step of the Installation Wizard.

Step 7. Setup validation

This is the required, final step of the Installation Wizard. At this step, KUMA creates a service resource set, and the Services are created automatically based on this set:

- The set of resources for the correlator is displayed under **Resources** → **Correlators**. It can be used to create new correlator services. When this set of resources changes, all services that operate based on this set of resources will start using the new parameters after the services restart. To do so, you can use the **Save and restart services** and **Save and update service configurations** buttons.

  A set of resources can be modified, copied, moved from one folder to another, deleted, imported, and exported, like other resources.

- Services are displayed in **Resources** → **Active services**. The services created using the Installation Wizard perform functions inside the KUMA program. To communicate with external parts of the network infrastructure, you need to install similar external services on the servers and assets intended for them. For example, an external correlator service should be installed on a server intended to process events, external storage services should be installed on servers with a deployed ClickHouse service, and external agent services should be installed on Windows assets that must both receive and forward Windows events.

To finish the Installation Wizard:

1. Click **Create and save service**.

   The **Setup validation** tab of the Installation Wizard displays a table of services created based on the set of resources selected in the Installation Wizard. The lower part of the window shows examples of commands that you must use to install external equivalents of these services on their intended servers and assets.

   For example:

/opt/kaspersky/kuma/kuma correlator --core https://kuma-example:<port used for communication with the KUMA Core> --id <service ID> --api.port <port used for communication with the service> --install

> The "kuma" file can be found inside the installer in the /kuma-ansible-installer/roles/kuma/files/ directory.

The port for communication with the KUMA Core, the service ID, and the port for communication with the service are added to the command automatically. You should also ensure the network connectivity of the KUMA system and open the ports used by its components if necessary.

2. Close the Wizard by clicking **Save**.

The correlator service is created in KUMA. Now the equivalent service must be installed on the server intended for processing events.

Installing a correlator in a KUMA network infrastructure

A correlator consists of two parts: one part is created inside the KUMA 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. Create the /opt/kaspersky/kuma/ folder.
3. Copy the "kuma" file to the /opt/kaspersky/kuma/ folder. The file is located in the installer in the /kuma-ansible-installer/roles/kuma/files/ folder.

   > Make sure the kuma file has sufficient rights to run.

4. Execute the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma correlator --core https://<KUMA Core server FQDN>:<port used by KUMA Core server for internal communication (port 7210 by default)> --id <service ID copied from the KUMA 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 collector

A collector consists of two parts: one part is created inside the KUMA web interface, and the other part is installed on a server in the network infrastructure intended for receiving 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 set of resources for a collector. Upon completion of the Wizard, the service itself is automatically created based on this set of resources.

To create a collector in the KUMA web interface,

Start the Collector Installation Wizard:

- In the KUMA web interface, in the **Resources** section, click **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.

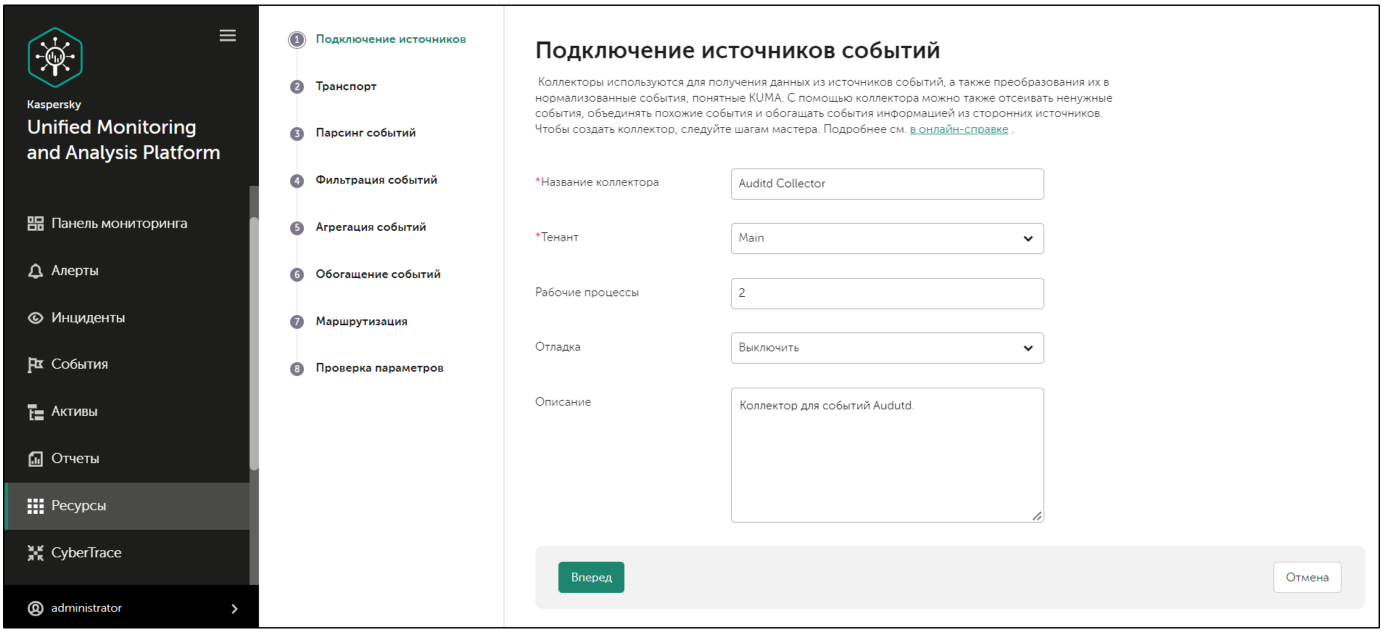
Starting the Collector Installation Wizard

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.



Подключение источников событий

Коллекторы используются для получения данных из источников событий, а также преобразования их в нормализованные события, понятные KUMA. С помощью коллектора можно также отсеивать ненужные события, объединять похожие события и обогащать события информацией из сторонних источников. Чтобы создать коллектор, следуйте шагам мастера. Подробнее см. в онлайн-справке .

*Название коллектора	Auditd Collector
*Тенант	Main
Рабочие процессы	2
Отладка	Выключить
Описание	Коллектор для событий Audutd.

Event Source Connection 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 basic settings of the collector:

1. In the **Collector name** field, enter a unique name for the service you are creating. The name must contain 1 to 128 Unicode characters.

   > When certain types of collectors are created, agents named "agent: <Collector name>, auto created" are also automatically created together with the collectors. If this type of agent was previously created and has not been deleted, it will be impossible to create a collector named <Collector name>. If this is the case, you will have to either specify a different name for the collector or delete the previously created agent.

2. In the **Tenant** drop-down list, select the tenant that will own the collector. The tenant selection determines what resources will be available when the collector is created.

   > If you return to this window from another subsequent step of the Installation Wizard and select another tenant, you will have to manually edit all the resources that you have added to the service. Only resources from the selected tenant and shared tenant can be added to the service.

3. If required, specify the number of processes that the service can run concurrently in the **Workers** field. By default, the number of worker processes is the same as the number of vCPUs on the server where the service

is installed.

4. If necessary, use the **Debug** 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 machine where the collector is installed, in the /opt/kaspersky/kuma/collector/<collector ID>/log/collector directory.

5. You can optionally add up to 256 Unicode characters describing the service in the **Description** field.

The main settings of the collector are specified. Proceed to the next step of the Installation Wizard.

Step 2. Transportation

This is a required step of the Installation Wizard. On the **Transport** tab of the Installation Wizard, select or create a connector and in its settings, specify the source of events for the collector service.

To add an existing connector to a resource set,

select the name of the required connector from the **Connector** drop-down list.

The **Transport** tab of the Installation Wizard displays the settings of the selected connector. You can open the selected connector for editing in a new browser tab 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:

   - tcp
   - udp
   - netflow
   - sflow
   - nats-jetstream
   - kafka
   - http
   - sql
   - file
   - ftp
   - nfs
   - wmi

- wec
- snmp

When using the **tcp** or **udp** connector type at the normalization stage, IP addresses of the assets from which the events were received will be written in the DeviceAddress event field if it is empty.

When using a **wmi** or **wec** connector, agents will be automatically created for receiving Windows events.

It is recommended to use the default encoding (UTF-8), and to apply other settings only if bit characters are received in the fields of events.

> Making KUMA collectors to listen on ports up to 1,000 requires running the service of the relevant collector with root privileges. To do this, after installing the collector, add the line `AmbientCapabilities = CAP_NET_BIND_SERVICE` to its systemd configuration file in the [Service] section.
> The systemd file is located in the /usr/lib/systemd/system/kuma-collector-<`collector ID`>.service directory.

The connector is added to the resource set of the collector. The created connector is only available in this resource set and is not displayed in the web interface **Resources** → **Connectors section**.

Proceed to the next step of the Installation Wizard.

Step 3. Event parsing

This is a required step of the Installation Wizard. On the **Event parsing** tab of the Installation Wizard, select or create a normalizer whose settings will define the rules for converting raw events into normalized events. You can add multiple event parsing rules to the normalizer to implement complex event processing logic. You can test the normalizer using test events.

When creating a new normalizer in the Installation Wizard, by default it is saved in the set of resources for the collector and cannot be used in other collectors. The **Save normalizer** check box lets you create the normalizer as a separate resource, in which case the normalizer can be selected in other collectors of the tenant.

> If, when changing the settings of a collector resource set, you change or delete conversions in a normalizer connected to it, the edits will not be saved, and the normalizer itself may be corrupted. If you need to modify conversions in a normalizer that is already part of a service, the changes must be made directly to the normalizer under **Resources** → **Normalizers** in the web interface.

Adding a normalizer

To add an existing normalizer to a resource set:

1. Click the **Add event parsing** button.

   This opens the **Basic event parsing** window with the normalizer settings and the **Normalization scheme** tab active.

2. In the **Normalizer** drop-down list, select the required normalizer. The drop-down list includes normalizers belonging to the tenant of the collector and the Shared tenant.

   The **Basic event parsing** window displays the settings of the selected normalizer.

   If you want to edit the normalizer settings, in the **Normalizer** drop-down list, click the pencil icon next to the name of the relevant normalizer. This opens the **Edit normalizer** window with a dark circle. Clicking the dark circle opens the **Basic event parsing** window where you can edit the normalizer settings.

   If you want to edit advanced parsing settings, move the cursor over the dark circle to make a plus icon appear; click the plus icon to open the **Advanced event parsing** window. For details about configuring advanced event parsing, see below.

3. Click **OK**.

The normalizer is displayed as a dark circle on the **Basic event parsing** tab of the Installation Wizard. Clicking on the circle will open the normalizer options for viewing.

To create a new normalizer in the collector:

1. At the Event parsing step, on the **Parsing schemes** tab, click the **Add event parsing**.

   This opens the **Basic event parsing** window with the normalizer settings and the **Normalization scheme** tab active.

2. If you want to save the normalizer as a separate resource, select the **Save normalizer** check box; this makes the saved normalizer available for use in other collectors of the tenant. This check box is cleared by default.

3. In the **Name** field, enter a unique name for the normalizer. The name must contain 1 to 128 Unicode characters.

4. In the **Parsing method** drop-down list, select the type of events to receive. Depending on your choice, you can use the preconfigured rules for matching event fields or set your own rules. When you select some of the parsing methods, additional settings fields may need to be filled.

   Available parsing methods:

   - json

     This parsing method is used to process JSON data where each object, including its nested objects, occupies a single line in a file.

     When processing files with hierarchically arranged data, you can access the fields of nested objects by specifying the names of the parameters dividing them by a period. 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 additional normalizer in its entirety.

     Newline characters can be `\n` and `\r\n`. Strings must be UTF-8 encoded.

     If you want to send the raw event for advanced normalization, at each nesting level in the **Advanced event parsing** window, select **Yes** in the **Keep raw event** drop-down list.

   - cef

This parsing method is used to process CEF data.

When choosing this method, you can use the preconfigured rules for converting events to the KUMA format by clicking the **Apply default mapping** button.

- regexp ?

This parsing method is used to create custom rules for processing data in a format using regular expressions.

In the **Normalization** parameter block field, add a regular expression (RE2 syntax) with named capture groups. The name of a group and its value will be interpreted as the field and the value of the raw event, which can be converted into an event field in KUMA format.

To add event handling rules:

1. Copy an example of the data you want to process to the **Event examples** field. This is an optional but recommended step.

2. In the **Normalization** parameter block field add a regular expression with named capture groups in RE2 syntax, for example "(?P<name>regexp)". The regular expression added to the **Normalization** parameter must exactly match the event. Also, when developing the regular expression, it is recommended to use special characters that match the starting and ending positions of the text: ^, $.

   You can add multiple regular expressions by using the **Add regular expression** button. If you need to remove the regular expression, use the ✕ button.

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. Now you can select the corresponding KUMA field in the column next to each capture group. Otherwise, if you named the capture groups in accordance with the CEF format, you can use the automatic CEF mapping by selecting the **Use CEF syntax for normalization** check box.

Event handling rules were added.

- syslog ?

This parsing method is used to process data in syslog format.

When choosing this method, you can use the preconfigured rules for converting events to the KUMA format by clicking the **Apply default mapping** button.

- csv ?

This parsing method is used to create custom rules for processing CSV data.

When choosing this 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.

- kv ?

This parsing method is used to process data in key-value pair format.

If you select this method, you must provide values in the following required fields:

- **Pair delimiter**—specify a character that will serve as a delimiter for key-value pairs. You can specify any one-character (1 byte) value, provided that the character does not match the value delimiter.
- **Value delimiter**—specify a character that will serve as a delimiter between the key and the value. You can specify any one-character (1 byte) value, provided that the character does not match the delimiter of key-value pairs.

- **xml** ?

This parsing method is used to process XML data in which each object, including its nested objects, occupies a single line in a file. Files are processed line by line.

If you want to send the raw event for advanced normalization, at each nesting level in the **Advanced event parsing** window, select **Yes** in the **Keep raw event** drop-down list.

When this method is selected in the parameter block **XML attributes** you can specify the key attributes to be extracted from tags. If an XML structure has several attributes with different values in the same tag, you can indicate the necessary value by specifying its key in the **Source** column of the **Mapping** table.

To add key XML attributes,

Click the **Add field** button, and in the window that appears, specify the path to the required attribute.

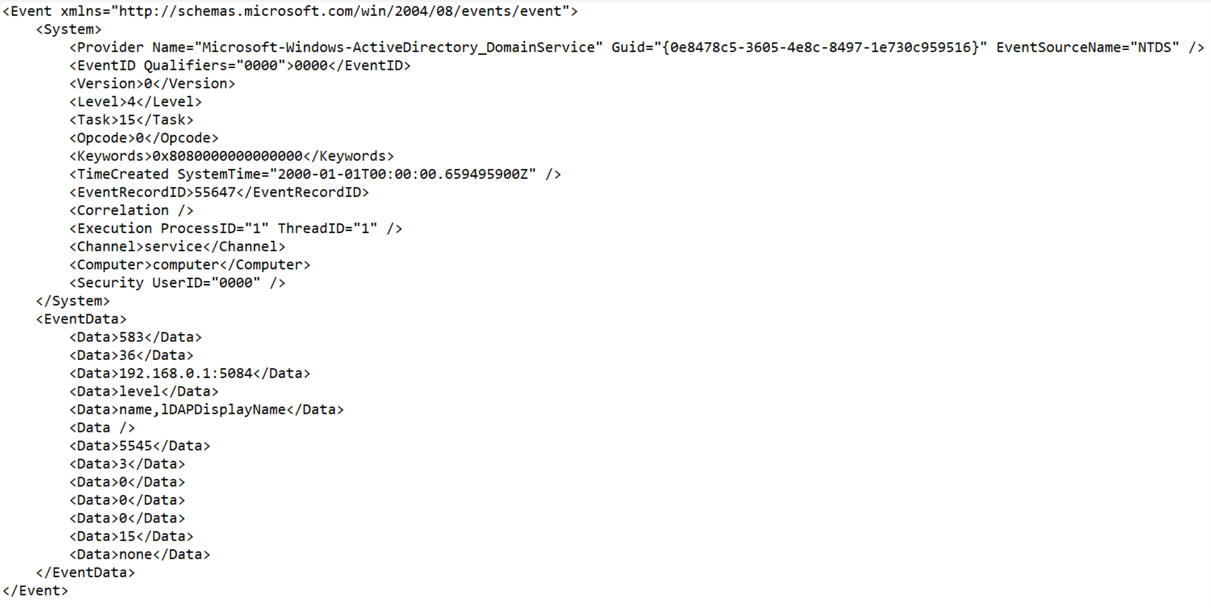
You can add more than one attribute. Attributes can be removed one at a time using the cross icon or all at once using the **Reset** button.

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

Tag numbering is available as of KUMA 2.1.3. This functionality allows automatically numbering tags in XML events, which lets you parse an event with identical tags or unnamed tags, such as <Data>.

As an example, we will use the **Tag numbering** functionality to number the tags of the EventData attribute of 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>0</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 such events, you must:

- Configure tag numbering.
- Configure data mapping for numbered tags with KUMA event fields.

Simultaneous use of **XML attributes** and **Tag numbering** leads to incorrect operation of the normalizer. If an attribute contains unnamed tags or the identical tags, we recommend using the **Tag numbering** functionality. If the attribute contains only named tags, use **XML attributes**.

To configure parsing of events with identically named or unnamed tags:

1. Create a new normalizer or open an existing normalizer for editing.

2. In the **Basic event parsing** window of the normalizer, in the **Parsing method** drop-down list, select 'xml' and in the **Tag numbering** field, click **Add field**.

   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 number to be assigned to a tag is 0. If the tag is empty, for example, <Data />, it is also assigned a number.

3. To configure data mapping, under **Mapping**, click **Add row** and do the following:

   a. In the new row, in the **Source** field, enter the full path to the tag and its index. For the Microsoft Windows event from the example above, the full path with indices look like this:

      - 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.

4. To save changes:

   - If you created a new normalizer, click **Save**.

   - If you edited an existing normalizer, click **Update configuration** in the collector to which the normalizer is linked.

Parsing is configured.

- netflow5 ?

This parsing method is used to process data in the NetFlow v5 format.

When choosing this method, you can use the preconfigured rules for converting events to the KUMA format by clicking the **Apply default mapping** button. If the netflow5 type is selected for the main parsing, extra normalization is not available.

> In mapping rules, the protocol type for **netflow5** is not indicated in the fields of KUMA events by default. 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.
>
> When choosing this method, you can use the preconfigured rules for converting events to the KUMA format by clicking the **Apply default mapping** button. If the netflow9 type is selected for the main parsing, extra normalization is not available.
>
> > In mapping rules, the protocol type for **netflow9** is not indicated in the fields of KUMA events by default. 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.
>
> When choosing this method, you can use the preconfigured rules for converting events to the KUMA format by clicking the **Apply default mapping** button. If the sflow5 type is selected for the main parsing, extra normalization is not available.

- ipfix

> This parsing method is used to process IPFIX data.
>
> When choosing this method, you can use the preconfigured rules for converting events to the KUMA format by clicking the **Apply default mapping** button. If the ipfix type is selected for the main parsing, extra normalization is not available.
>
> > In mapping rules, the protocol type for **ipfix** is not indicated in the fields of KUMA events by default. When parsing data in NetFlow format, on the **Enrichment** normalizer tab, you must create a **constant** data enrichment rule that adds the `netflow` value to the `DeviceProduct` target field.

- sql—this method becomes available only when using a sql type connector.

> The normalizer uses this method to process data obtained by making a selection from the database.

5. In the **Keep raw event** drop-down list, specify whether to store the original raw event in the newly created normalized event. Available values:

   - **Don't save**—do not save the raw event. This is the default setting.
   - **Only errors**—save the raw event in the `Raw` field of the normalized event if errors occurred when parsing it. This value is convenient to use when debugging a service. In this case, every time an event has a non-empty `Raw` field, you know there was a problem.
   - **Always**—always save the raw event in the `Raw` field of the normalized event.

6. In the **Keep extra fields** drop-down list, choose whether you want to store the raw event fields in the normalized event if no mapping rules have been configured for them (see below). The data is stored in the Extra event field. By default, fields are not saved.

7. Copy an example of the data you want to process to the **Event examples** field. This is an optional but recommended step.

8. In the **Mapping** table, configure the mapping of raw event fields to event fields in the KUMA format:

   a. In the **Source** column, provide the name of the raw event field that you want to convert into the KUMA event field.

      For details about the field format, refer to the Normalized event data model article. For a description of the mapping, refer to the Mapping fields of predefined normalizers article.

      Clicking the 🔧 button next to the field names in the **Source** column opens the **Conversion** window, in which you can use the **Add conversion** button to create rules for modifying the original data before they are written to the KUMA event fields.

      Available conversions

Conversions are changes that can be applied to a value before it gets written to the event field. The conversion type is selected from a drop-down list.

Available conversions:

- **lower**—is used to make all characters of the value lowercase
- **upper**—is used to make all characters of the value uppercase
- **regexp** – used to convert a value using the regular expression RE2. When this conversion type is selected, the field appears where regular expression should be added.
- **substring**—is used to extract characters in the position range specified in the **Start** and **End** fields. These fields appear when this conversion type is selected.
- **replace**—is used to replace specified character sequence with the other character sequence. When this type of conversion is selected, new fields appear:
  - **Replace chars**—in this field you can specify the character sequence that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- **trim**—used to simultaneously remove the characters specified in the **Chars** field from the leading and end positions of the value. The field appears when this type of conversion is selected. For example, a **trim** conversion with the `Micromon` value applied to `Microsoft-Windows-Sysmon` results in `soft-Windows-Sys`.
- **append** is used to add the characters specified in the **Constant** field to the end of the event field value. The field appears when this type of conversion is selected.
- **prepend**—used to prepend the characters specified in the **Constant** field to the start of the event field value. The field appears when this type of conversion is selected.
- **replace with regexp**—is used to replace RE2 regular expression results with the character sequence.
  - **Expression**—in this field you can specify the regular expression which results that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, such a string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the "String" type, all types of conversions are available.
- For fields of the "Number" and "Float" types, the following types of conversions are available: regexp, substring, replace, trim, append, prepend, replaceWithRegexp, decodeHexString, decodeBase64String, decodeBase64URLString.
- For fields of "Array of strings", "Array of numbers", and "Array of floats" types, the following types of conversions are available: append, prepend.

In the **Conversion** window, you can swap the added rules by dragging them by the ⋮⋮ icon; you can also delete them using the × icon.

b. In the **KUMA field** column, select the required KUMA event field from the drop-down list. You can search for fields by entering their names in the field.

c. If the name of the KUMA event field selected at the previous step begins with `DeviceCustom*` or `Flex*`, you can add a unique custom label in the **Label** field.

New table rows can be added by using the **Add row** button. Rows can be deleted individually using the × button or all at once using the **Clear all** button.

If you want KUMA to enrich events with asset information, and the asset information to be available in the alert card when a correlation rule is triggered, in the **Mapping** table, configure a mapping of host address and host name fields depending on the purpose of the asset. For example, the mapping can apply to SourceAddress and SourceHostName, or DestinationAddress and DestinationHostName fields. As a result of enrichment, the event card includes a SourceAssetID or DestinationAssetID field, and a link to the asset card. Also, as a result of enrichment, asset information is available in the alert card.

If you have loaded data into the **Event examples** field, the table will have an **Examples** column containing examples of values carried over from the raw event field to the KUMA event field.

9. Click **OK**.

The normalizer is displayed as a dark circle on the **Event parsing** tab of the Installation Wizard. If you want to open the normalizer settings for viewing, click the dark circle. When you hover the mouse over the circle, a plus sign is displayed. Click it to add event parsing rules (see below).

Enriching normalized events with additional data

You can add additional data to newly created normalized events by creating enrichment rules in the normalizer. These enrichment rules are stored in the normalizer where they were created. There can be more than one enrichment rule.

To add enrichment rules to the normalizer:

1. Select the main or additional normalization rule to open a window, and in that window, click the **Enrichment** tab.
2. Click the **Add enrichment** button.

   The enrichment rule parameter block appears. You can delete the group of settings using the × button.

3. Select the enrichment type from the **Source kind** drop-down list. Depending on the selected type, you may see advanced settings that will also need to be completed.

   Available Enrichment rule source types:

   - constant ?

     This type of enrichment is used when a constant needs to be added to an event field. Settings of this type of enrichment:

     - In the **Constant** field, specify the value that should be added to the event field. The value may not be longer than 255 Unicode characters. If you leave this field blank, the existing event field value will be cleared.
     - In the **Target field** drop-down list, select the KUMA event field to which you want to write 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.

     When this type is selected in the **Dictionary name** drop-down list, you must select the dictionary that will provide the values. In the **Key fields** settings block, you must use the **Add field** button to select the event fields whose values will be used for dictionary entry selection.

     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 3 elements: "a", "b" and "c". The following value is 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 two fields: the SA.StringArrayOne extended schema field and the Code field. The SA.StringArrayOne extended schema field contains 3 elements: "a", "b", and "c"; the Code string field contains the character sequence "myCode". The following value is passed to the dictionary as the key: ['a','b','c']|myCode.

   - table ?

This type of enrichment is used if you need to add a value from the dictionary of the **Table** type.

When this enrichment type is selected in the **Dictionary name** drop-down list, select the dictionary for providing the values. In the **Key fields** group of settings, use the **Add field** button to select the event fields whose values are used for dictionary entry selection.

In the **Mapping** table, configure the dictionary fields to provide data and the event fields to receive data:

- In the **Dictionary field** column, select the dictionary field. The available fields depend on the selected dictionary resource.

- In the **KUMA field** column, select the event field to which the value is written. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written to them.

New table rows can be added by using the **Add new element** button. Columns can be deleted using the ✕ button.

- event

This type of enrichment is used when you need to write a value from another event field to the current event field. Settings of this type of enrichment:

- In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.
- In the **Source field** drop-down list, select the event field whose value will be written to the target field.
- Clicking the 🔧 button opens the **Conversion** window in which you can, using the **Add conversion** button, create rules for modifying the original data before writing them to the KUMA event fields.

  Available conversions

Conversions are changes that can be applied to a value before it gets written to the event field. The conversion type is selected from a drop-down list.

Available conversions:

- **lower**—is used to make all characters of the value lowercase
- **upper**—is used to make all characters of the value uppercase
- **regexp** – used to convert a value using the regular expression RE2. When this conversion type is selected, the field appears where regular expression should be added.
- **substring**—is used to extract characters in the position range specified in the **Start** and **End** fields. These fields appear when this conversion type is selected.
- **replace**—is used to replace specified character sequence with the other character sequence. When this type of conversion is selected, new fields appear:
  - **Replace chars**—in this field you can specify the character sequence that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- **trim**—used to simultaneously remove the characters specified in the **Chars** field from the leading and end positions of the value. The field appears when this type of conversion is selected. For example, a **trim** conversion with the `Micromon` value applied to `Microsoft-Windows-Sysmon` results in `soft-Windows-Sys`.
- **append** is used to add the characters specified in the **Constant** field to the end of the event field value. The field appears when this type of conversion is selected.
- **prepend**—used to prepend the characters specified in the **Constant** field to the start of the event field value. The field appears when this type of conversion is selected.
- **replace with regexp**—is used to replace RE2 regular expression results with the character sequence.
  - **Expression**—in this field you can specify the regular expression which results that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, such a string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the "String" type, all types of conversions are available.
- For fields of the "Number" and "Float" types, the following types of conversions are available: regexp, substring, replace, trim, append, prepend, replaceWithRegexp, decodeHexString, decodeBase64String, decodeBase64URLString.
- For fields of "Array of strings", "Array of numbers", and "Array of floats" types, the following types of conversions are available: append, prepend.

When using enrichment of events that have the "Event" selected as the "Source kind" setting and the fields of the extended event schema are used as arguments, the following special considerations apply:

- If the source field is an "Array of strings" field and the target field is a "String" field, the values are written to the target field in the 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 operation is written to the DeviceCustomString1 event schema field. As a result of the operation, the DeviceCustomString1 field contains ["string1", "string2", "string3"].

- If the source field is an "Array of strings" field and the target field is an "Array of strings" field, the values are written to the target field, and commas (",") are used as the separator character.

  Example: The SA.StringArrayOne extended event schema field contains values: "string1", "string2", "string3". An event enrichment operation is performed. The result of the operation is written to the SA.StringArrayTwo event schema field. As a result of the operation, the SA.StringArrayTwo field contains "string1", "string2", "string3".

- **template**

This type of enrichment is used when you need to write a value obtained by processing Go templates into the event field. Settings of this type of enrichment:

- Put the Go template into the **Template** field.

  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.

  Example: `Attack on {{.DestinationAddress}} from {{.SourceAddress}}`.

- In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.

If you are using enrichment of events that have the "Template" type selected as the "Source kind" setting, 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.

Example:

```
{{.SA.StringArrayOne}}
```

Example:

```
{{- range $index, $element := . SA.StringArrayOne -}}

{{- if $index}}, {{end}}"{{$element}}"{{- end -}}
```

4. In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.

   This setting is not available for the enrichment source of the **Table** type.

5. If you want to enable details in the normalizer log, set the **Debug** toggle switch to enabled. Details are disabled by default.

6. Click **OK**.

Event enrichment rules with the additional data are added to the normalizer, to the selected parsing rule.

Configuring parsing linked to IP addresses

You can direct events from multiple IP addresses, from sources of different types, to the same collector, and the collector will apply the corresponding configured normalizers.

You can use this method for collectors with a connector of the UDP, TCP, or HTTP type. If a UDP, TCP, or HTTP connector is specified in the collector at the **Transport** step, then at the **Event parsing** step, you can specify multiple IP addresses on the **Parsing settings** tab and choose the normalizer that you want to use for events coming from the specified addresses. The following types of normalizers are available: json, cef, regexp, syslog, csv, kv, xml.

In a collector with configured normalizers linked to IP addresses, if you change the connector type to any type other than UDP, TCP, HTTP, the **Parsing settings** tab disappears and only the first of the previously specified normalizers is specified at the **Parsing** step. The tab disappears from the web interface immediately, but the changes are applied after the resource is saved. If you want to restore the previous settings, exit the collector installation wizard without saving.

For normalizers of the Syslog and regexp types, you can use a normalizer chain by specifying extra normalization conditions depending on the value of the DeviceProcessName field. The difference from extra normalization is that you can specify shared normalizers.

To configure parsing with linking to IP addresses:

1. At the **Event parsing** step, go to the **Parsing settings** tab.

2. In the **IP address(-es)** field, specify one or more IP addresses from which events will be received. You can specify multiple IP addresses separated by commas. Available format: IPv4. The length of the address list is unlimited; however, we recommend specifying a reasonable number of addresses to keep the load on the collector balanced. This field is mandatory if you want to apply multiple normalizers in one collector.

   Limitation: for each IP+normalizer combination, the IP address must be unique. KUMA checks the uniqueness of addresses, and if you specify the same IP address for different normalizers, the "The field must be unique" message is displayed.

   If you want to send all events to the same normalizer without specifying IP addresses, we recommend creating a separate collector. We also recommend creating a separate collector with one normalizer if you want to apply the same normalizer to events from a large number of IP addresses; this helps improve the performance.

3. In the **Normalizer** field, create a normalizer or select an existing normalizer from the drop-down list. The arrow next to the drop-down list takes you to the **Parsing schemes** tab.

   Normalization is triggered if you have a connector type configured: UDP, TCP, HTTP; the event source header must be specified in the HTTP case.

   Taking into account the available connectors, the following normalizer types are available for automatic source recognition: json, cef, regexp, syslog, csv, kv, xml.

4. If you selected the Syslog or regexp normalizer type, you can **Additional condition**. Conditional normalization is available if **Field mapping** for DeviceProcessName is configured in the main normalizer. Under **Condition**, specify the process name in the DeviceProcessName field and create a normalizer or select an existing normalizer from the drop-down list. You can specify multiple combinations of DeviceProcessName + normalizer, normalization is performed until the first match is achieved.

Parsing with linking to IP addresses is configured.

Creating a structure of event normalization rules

To implement a complex event processing logic, you can add multiple event parsing rules to the normalizer. Events are transmitted between the parsing rules depending on the specified conditions. The sequence of creating parsing rules is important. The event is processed sequentially, and its path is shown using arrows.

To create an additional parsing rule:

1. Create a normalizer (see above).

   The created normalizer is displayed in the window as a dark circle.

2. Hover the mouse over the circle and click the plus sign button that appears.

3. In the **Additional event parsing** window that opens, specify the parameters of the additional event parsing rule:

   - **Extra normalization conditions** tab:

     If you want to send a raw event for extra normalization, select **Yes** in the **Keep raw event** drop-down list. The default value is **No**. We recommend passing a raw event to normalizers of json and xml types. If you want to send a raw event for extra normalization to the second, third, etc nesting levels, at each nesting level, select **Yes** in the **Keep raw event** drop-down list.

  To send only the events with a specific field to the additional normalizer, specify this field in the **Field to pass into normalizer** field.

  On this tab, you can also define other conditions. When these conditions are met, the event is sent for additional parsing.

- **Normalization scheme** tab:

  On this tab, you can configure event processing rules, similar to the main normalizer settings (see above). The **Keep raw event** setting is not available. The **Event examples** field displays the values specified when the initial normalizer was created.

- **Enrichment** tab:

  On this tab, you can configure event enrichment rules (see above).

4. Click **OK**.

The additional parsing rule is added to the normalizer. It is displayed as a dark block with the conditions under which this rule is triggered. You can change the settings of the additional parsing rule by clicking it. If you hover the mouse over the additional parsing rule, a plus button appears. You can use this button to create a new additional parsing rule. To delete a normalizer, use the button with the trash icon.

The upper right corner of the window contains a search window where you can search parsing rules by name.

Proceed to the next step of the Installation Wizard.

Step 4. Filtering events

This is an optional step of the Installation Wizard. The **Event filtering** tab of the Installation Wizard allows you to select or create a filter whose settings specify the conditions for selecting events. You can add multiple filters to the collector. You can swap the filters by dragging them by the ⠿ icon as well as delete them. Filters are combined by the AND operator.

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:

   - The **Add condition** button is used to add filtering conditions. You can select two values (two operands, left and right) and assign the operation you want to perform with the selected values. The result of the operation is either True or False.

- In the **operator** drop-down list, select the function to be performed by the filter.

  In this drop-down list, you can select the **do not match case** check box if the operator should ignore the case of values. This check box is ignored if the **InSubnet**, **InActiveList**, **InCategory**, and **InActiveDirectoryGroup** operators are selected. This check box is cleared by default.

  Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

    - In the **Left operand** and **Right operand** drop-down lists, select where the data to be filtered will come from. As a result of the selection, Advanced settings will appear. Use them to determine the exact value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

    - You can use the **If** drop-down list to choose whether you need to create a negative filter condition.

  Conditions can be deleted using the × button.

- The **Add group** button is used to add groups of conditions. Operator **AND** can be switched between **AND**, **OR**, and **NOT** values.

  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. 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 ?

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. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

   The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

   This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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.

Rule-based enrichment

There can be more than one enrichment rule. You can add them by clicking the **Add enrichment** button and can remove them by clicking the ✕ button. You can use existing enrichment rules or create rules directly in the Installation Wizard.

To add an existing enrichment rule to a set of resources:

1. Click **Add enrichment**.

   This opens the enrichment rules settings block.

2. In the **Enrichment rule** drop-down list, select the relevant resource.

The enrichment rule is added to the set of resources for the collector.

To create a new enrichment rule in a set of resources:

1. Click **Add enrichment**.

   This opens the enrichment rules settings block.

2. In the **Enrichment rule** drop-down list, select **Create new**.

3. In the **Source kind** drop-down list, select the source of data for enrichment and define its corresponding settings:

- constant

  This type of enrichment is used when a constant needs to be added to an event field. Settings of this type of enrichment:

  - In the **Constant** field, specify the value that should be added to the event field. The value may not be longer than 255 Unicode characters. If you leave this field blank, the existing event field value will be cleared.

  - In the **Target field** drop-down list, select the KUMA event field to which you want to write 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.

  When this type is selected in the **Dictionary name** drop-down list, you must select the dictionary that will provide the values. In the **Key fields** settings block, you must use the **Add field** button to select the event fields whose values will be used for dictionary entry selection.

  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 3 elements: "a", "b" and "c". The following value is 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 two fields: the SA.StringArrayOne extended schema field and the Code field. The SA.StringArrayOne extended schema field contains 3 elements: "a", "b", and "c"; the Code string field contains the character sequence "myCode". The following value is passed to the dictionary as the key: ['a','b','c']|myCode.

- event

This type of enrichment is used when you need to write a value from another event field to the current event field. Settings of this type of enrichment:

- In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.
- In the **Source field** drop-down list, select the event field whose value will be written to the target field.
- In the **Conversion** settings block, you can create rules for modifying the original data before it is written to the KUMA event fields. The conversion type can be selected from the drop-down list. You can use the **Add conversion** and **Delete** buttons to add or delete a conversion, respectively. The order of conversions is important.

  Available conversions ?

Conversions are changes that can be applied to a value before it gets written to the event field. The conversion type is selected from a drop-down list.

Available conversions:

- **lower**—is used to make all characters of the value lowercase
- **upper**—is used to make all characters of the value uppercase
- **regexp** – used to convert a value using the regular expression RE2. When this conversion type is selected, the field appears where regular expression should be added.
- **substring**—is used to extract characters in the position range specified in the **Start** and **End** fields. These fields appear when this conversion type is selected.
- **replace**—is used to replace specified character sequence with the other character sequence. When this type of conversion is selected, new fields appear:
  - **Replace chars**—in this field you can specify the character sequence that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- **trim**—used to simultaneously remove the characters specified in the **Chars** field from the leading and end positions of the value. The field appears when this type of conversion is selected. For example, a **trim** conversion with the `Micromon` value applied to `Microsoft-Windows-Sysmon` results in `soft-Windows-Sys`.
- **append** is used to add the characters specified in the **Constant** field to the end of the event field value. The field appears when this type of conversion is selected.
- **prepend**—used to prepend the characters specified in the **Constant** field to the start of the event field value. The field appears when this type of conversion is selected.
- **replace with regexp**—is used to replace RE2 regular expression results with the character sequence.
  - **Expression**—in this field you can specify the regular expression which results that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, such a string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the "String" type, all types of conversions are available.
- For fields of the "Number" and "Float" types, the following types of conversions are available: regexp, substring, replace, trim, append, prepend, replaceWithRegexp, decodeHexString, decodeBase64String, decodeBase64URLString.
- For fields of "Array of strings", "Array of numbers", and "Array of floats" types, the following types of conversions are available: append, prepend.

- template

  This type of enrichment is used when you need to write a value obtained by processing Go templates into the event field. Settings of this type of enrichment:

  - Put the Go template into the **Template** field.

    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.

    Example: `Attack on {{.DestinationAddress}} from {{.SourceAddress}}`.

  - In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.

  If you are using enrichment of events that have the "Template" type selected as the "Source kind" setting, 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.

  Example:

  {{.SA.StringArrayOne}}

  Example:

  {{- range $index, $element := . SA.StringArrayOne -}}

  {{- if $index}}, {{end}}"{{$element}}"{{- end -}}

- dns

This type of enrichment is used to send requests to a private network DNS server to convert IP addresses into domain names or vice versa. IP addresses are converted to DNS names only for private addresses: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 100.64.0.0/10.

Available settings:

- **URL**—in this field, you can specify the URL of a DNS server to which you want to send requests. You can use the **Add URL** button to specify multiple URLs.
- **RPS**—maximum number of requests sent to the server per second. The default value is `1,000`.
- **Workers**—maximum number of requests per one point in time. The default value is `1`.
- **Max tasks**—maximum number of simultaneously fulfilled requests. By default, this value is equal to the number of vCPUs of the KUMA Core server.
- **Cache TTL**—the lifetime of the values stored in the cache. The default value is `60`.
- **Cache disabled**—you can use this drop-down list to enable or disable caching. Caching is enabled by default.

- cybertrace

  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.
  - **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`.
  - **Mapping** (required)—this settings block contains the mapping table for mapping KUMA event fields to CyberTrace indicator types. The **KUMA field** column shows the names of KUMA event fields, and the **CyberTrace indicator** column shows the types of CyberTrace indicators.

    Available types of CyberTrace indicators:

    - ip
    - url
    - hash

    In the mapping table, you must provide at least one string. You can use the **Add row** button to add a string, and can use the × button to remove a string.

- timezone

This type of enrichment is used in collectors and correlators to assign a specific timezone to an event. Timezone information may be useful when searching for events that occurred at unusual times, such as nighttime.

When this type of enrichment is selected, the required timezone must be selected from the **Timezone** drop-down list.

Make sure that the required time zone is set on the server hosting the enrichment-utilizing service. For example, you can do this by using the `timedatectl list-timezones` command, which shows all time zones that are set on the server. For more details on setting time zones, please refer to your operating system documentation.

When an event is enriched, the time offset of the selected timezone relative to Coordinated Universal Time (UTC) is written to the DeviceTimeZone event field in the `+-hh:mm` format. For example, if you select the **Asia/Yekaterinburg** timezone, the value `+05:00` will be written to the DeviceTimeZone field. If the enriched event already has a value in the DeviceTimeZone field, it will be overwritten.

By default, if the timezone is not specified in the event being processed and enrichment rules by timezone are not configured, the event is assigned the timezone of the server hosting the service (collector or correlator) that processes the event. If the server time is changed, the service must be restarted.

Permissible time formats when enriching the DeviceTimeZone field

When processing incoming raw events in the collector, the following time formats can be automatically converted to the +-hh:mm format:

Time format in a processed event	Example
+-hh:mm	-07:00
+-hhmm	-0700
+-hh	-07

If the date format in the `DeviceTimeZone` field differs from the formats listed above, the collector server timezone is written to the field when an event is enriched with timezone information. You can create custom normalization rules for non-standard time formats.

- geographic data

This type of enrichment is used to add IP address geographic data to event fields. Learn more about linking IP addresses to geographic data.

When this type is selected, in the **Mapping geographic data to event fields** settings block, you must specify from which event field the IP address will be read, select the required attributes of geographic data, and define the event fields in which geographic data will be written:

1. In the **Event field with IP address** drop-down list, select the event field from which the IP address is read. Geographic data uploaded to KUMA is matched against this IP address.

   You can use the **Add event field with IP address** button to specify multiple event fields with IP addresses that require geographic data enrichment. You can delete event fields added in this way by clicking the **Delete event field with IP address** button.

   > When the `SourceAddress`, `DestinationAddress`, and `DeviceAddress` event fields are selected, the **Apply default mapping** button becomes available. You can use this button to add preconfigured mapping pairs of geographic data attributes and event fields.

2. For each event field you need to read the IP address from, select the type of geographic data and the event field to which the geographic data should be written.

   You can use the **Add geodata attribute** button to add field pairs for **Geodata attribute** – **Event field to write to**. You can also configure different types of geographic data for one IP address to be written to different event fields. To delete a field pair, click ×.

   - In the **Geodata attribute** field, select which geographic data corresponding to the read IP address should be written to the event. Available geographic data attributes: **Country**, **Region**, **City**, **Longitude**, **Latitude**.

   - In the **Event field to write to**, select the event field which the selected geographic data attribute must be written to.

   You can write identical geographic data attributes to different event fields. If you configure multiple geographic data attributes to be written to the same event field, the event will be enriched with the last mapping in the sequence.

4. Use the **Debug** 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 by the enrichment rule resource. You can select an existing filter from the drop-down list or **create** a new filter.

   Creating a filter in resources

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. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search condition (and, or, not) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button.

   You can view the nested filter settings by clicking the ↗ button.

The new enrichment rule was added to the set of resources for the collector.

LDAP enrichment

To enable enrichment using LDAP:

1. Click **Add enrichment with LDAP data**.

   This opens the settings block for LDAP enrichment.

2. In the **LDAP accounts mapping** settings block, use the **New domain** button to specify the domain of the user accounts. You can specify multiple domains.

3. In the **LDAP mapping** table, define the rules for mapping KUMA fields to LDAP attributes:

   - In the **KUMA field** column, indicate the KUMA event field which data should be compared to LDAP attribute.

   - In the **LDAP attribute** column, specify the attribute that must be compared with the KUMA event field. The drop-down list contains standard attributes and can be augmented with **custom attributes** ?.

Before configuring event enrichment using custom attributes, make sure that custom attributes are configured in AD.

To enrich events with accounts using custom attributes:

1. Add **Custom AD Account Attributes** in the LDAP connection settings.

   Standard **imported attributes from AD** cannot be added as custom attributes. For example, if you add the standard `accountExpires` attribute as a custom attribute, KUMA returns an error when saving the connection settings.

The following account attributes can be requested from Active Directory:

- `accountExpires`
- `badPasswordTime`
- `cn`
- `co`
- `company`
- `department`
- `description`
- `displayName`
- `distinguishedName`
- `division`
- `employeeID`
- `givenName`
- `l`
- `lastLogon`
- `lastLogonTimestamp`
- `Mail`
- `mailNickname`
- `managedObjects`
- `manager`
- `memberOf` (this attribute can be used for search during correlation)
- `mobile`
- `name`
- `objectCategory`
- `objectGUID` (this attribute always requested from Active Directory even if a user doesn't specify it)
- `objectSID`
- `physicalDeliveryOfficeName`

- `pwdLastSet`
- `sAMAccountName`
- `sAMAccountType`
- `sn`
- `streetAddress`
- `telephoneNumber`
- `title`
- `userAccountControl`
- `UserPrincipalName`
- `whenChanged`
- `whenCreated`

After you add custom attributes in the LDAP connection settings, the **LDAP attribute to receive** drop-down list in the collector automatically includes the new attributes. Custom attributes are identified by a question mark next to the attribute name. If you added the same attribute for multiple domains, the attribute is listed only once in the drop-down list. You can view the domains by moving your cursor over the question mark. Domain names are displayed as links. If you click a link, the domain is automatically added to **LDAP accounts mapping** if it was not previously added.

If you deleted a custom attribute in the LDAP connection settings, manually delete the row containing the attribute from the mapping table in the collector. Account attribute information in KUMA is updated each time you import accounts.

2. Import accounts.
3. In the collector, in the **LDAP mapping** table, define the rules for mapping KUMA fields to LDAP attributes.
4. Restart the collector.

   After the collector is restarted, KUMA begins enriching events with accounts.

- In the **KUMA event field to write to** column, specify in which field of the KUMA event the ID of the user account imported from LDAP should be placed if the mapping was successful.

You can use the **Add row** button to add a string to the table, and can use the × button to remove a string. You can use the **Apply default mapping** button to fill the mapping table with standard values.

Event enrichment rules for data received from LDAP were added to the group of resources for the collector.

If you add an enrichment to an existing collector using LDAP or change the enrichment settings, you must stop and restart the service.

Proceed to the next step of the Installation Wizard.

Step 7. Routing

This is an optional step of the Installation Wizard. On the **Routing** tab of the Installation Wizard, you can select or create destinations with settings indicating the forwarding destination of events processed by the collector. Typically, events from the collector are routed to two points: to the correlator to analyze and search for threats; and to the storage, both for storage and so that processed events can be viewed later. Events can be sent to other locations as needed. There can be more than one destination point.

To add an existing destination to a collector resource set:

1. In the **Add destination** drop-down list, select the type of destination resource you want to add:

   - Select **Storage** if you want to configure forwarding of processed events to the storage.
   - Select **Correlator** if you want to configure forwarding of processed events to a correlator.
   - Select **Other** if you want to send events to other locations.

     > This type of resource includes correlator and storage services that were created in previous versions of the program.

   The **Add destination** window opens where you can specify parameters for events forwarding.

2. In the **Destination** drop-down list, select the necessary destination.

   The window name changes to **Edit destination**, and it displays the settings of the selected resource. To open the settings of a destination for editing in a new browser tab, click ↗.

3. Click **Save**.

The selected destination is displayed on the Installation Wizard tab. A destination resource can be removed from the resource set by selecting it and clicking **Delete** in the opened window.

To add a new destination resource to a collector resource set:

1. In the **Add destination** drop-down list, select the type of destination resource you want to add:

   - Select **Storage** if you want to configure forwarding of processed events to the storage.
   - Select **Correlator** if you want to configure forwarding of processed events to a correlator.
   - Select **Other** if you want to send events to other locations.

     > This type of resource includes correlator and storage services that were created in previous versions of the program.

   The **Add destination** window opens where you can specify parameters for events forwarding.

2. Specify the settings on the **Basic settings** tab:

   - In the **Destination** drop-down list, select **Create new**.

- In the **Name** field, enter a unique name for the destination resource. The name must contain 1 to 128 Unicode characters.
- Use the **Disabled** toggle button to specify whether events will be sent to this destination. By default, sending events is enabled.
- Select the **Type** for the destination resource:
  - Select **storage** if you want to configure forwarding of processed events to the storage.
  - Select **correlator** if you want to configure forwarding of processed events to a correlator.
  - Select **nats-jetstream**, **tcp**, **http**, **kafka**, or **file** if you want to configure sending events to other locations.
- Specify the **URL** to which events should be sent in the hostname:<API port> format.

  You can specify multiple destination addresses using the **URL** button for all types except **nats-jetstream**, **file**, and **diode**.
- For the **nats-jetstream** and **kafka** types, use the **Topic** field to specify which topic the data should be written to. The topic must contain Unicode characters. The Kafka topic is limited to 255 characters.

3. If necessary, specify the settings on the **Advanced settings** tab. The available settings vary based on the selected destination resource type:

- **Compression** is a drop-down list where you can enable Snappy compression. By default, compression is **disabled**.
- **Proxy** is a drop-down list for proxy server selection.
- The **Buffer size** field is used to set buffer size (in bytes) for the destination. The default value is 1 MB, and the maximum value is 64 MB.
- **Timeout** field is used to set the timeout (in seconds) for another service or component response. The default value is `30`.
- **Disk buffer size limit** field is used to specify the size of the disk buffer in bytes. The default size is 10 GB.
- **Cluster ID** is the ID of the NATS cluster.
- **TLS mode** is a drop-down list where you can specify the conditions for using TLS encryption:
  - **Disabled** (default)—do not use TLS encryption.
  - **Enabled**—encryption is enabled, but without verification.
  - **With verification**—use encryption with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during program installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/.

  When using TLS, it is impossible to specify an IP address as a URL.
- **URL selection policy** is a drop-down list in which you can select a method for determining which URL to send events to if several URLs have been specified:

  - **Any**. Events are sent to one of the available URLs as long as this URL receives events. If the connection is broken (for example, the receiving node is disconnected) a different URL will be selected as the events destination.

  - **Prefer first**. Events are sent to the first URL in the list of added addresses. If it becomes unavailable, events are sent to the next available node in sequence. When the first URL becomes available again, events start to be sent to it again.

  - **Balanced** means that packages with events are evenly distributed among the available URLs from the list. Because packets are sent either on a destination buffer overflow or on the flush timer, this URL selection policy does not guarantee an equal distribution of events to destinations.

- **Delimiter** is used to specify the character delimiting the events. By default, `\n` is used.

- **Path**—the file path if the **file** destination type is selected.

- **Buffer flush interval**—this field is used to set the time interval (in seconds) at which the data is sent to the destination. The default value is `100`.

- **Workers**—this field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.

- You can set health checks using the **Health check path** and **Health check timeout** fields. You can also disable health checks by selecting the **Health Check Disabled** check box.

- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

- The **Disk buffer disabled** drop-down list is used to enable or disable the use of a disk buffer. By default, the disk buffer is disabled.

  The disk buffer is used if the collector cannot send normalized events to the destination. The amount of allocated disk space is limited by the value of the **Disk buffer size limit** setting.

  If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer.

- In the **Filter** section, you can specify the conditions to define events that will be processed by this resource. You can select an existing filter from the drop-down list or **create** a new filter.

  Creating a filter in resources

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. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search condition (and, or, not) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button.

   You can view the nested filter settings by clicking the ⧉ button.

4. Click **Save**.

   The created destination is displayed on the Installation Wizard tab. A destination resource can be removed from the resource set by selecting it and clicking **Delete** in the opened window.

Proceed to the next step of the Installation Wizard.

Step 8. Setup validation

This is the required, final step of the Installation Wizard. At this step, KUMA creates a service resource set, and the Services are created automatically based on this set:

- The set of resources for the collector is displayed under **Resources** → **Collectors**. It can be used to create new collector services. When this set of resources changes, all services that operate based on this set of resources will start using the new parameters after the services restart. To do so, you can use the **Save and restart services** and **Save and update service configurations** buttons.

  A set of resources can be modified, copied, moved from one folder to another, deleted, imported, and exported, like other resources.

- Services are displayed in **Resources** → **Active services**. The services created using the Installation Wizard perform functions inside the KUMA program. To communicate with external parts of the network infrastructure, you need to install similar external services on the servers and assets intended for them. For example, an external collector service should be installed on a server intended as an events recipient, external storage services should be installed on servers that have a deployed ClickHouse service, and external agent services should be installed on the Windows assets that must both receive and forward Windows events.

To finish the Installation Wizard:

1. Click **Create and save service**.

   The **Setup validation** tab of the Installation Wizard displays a table of services created based on the set of resources selected in the Installation Wizard. The lower part of the window shows examples of commands that you must use to install external equivalents of these services on their intended servers and assets.

For example:

/opt/kaspersky/kuma/kuma collector --core https://kuma-example:<port used for communication with the KUMA Core> --id <service ID> --api.port <port used for communication with the service> --install

> The "kuma" file can be found inside the installer in the /kuma-ansible-installer/roles/kuma/files/ directory.

The port for communication with the KUMA Core, the service ID, and the port for communication with the service are added to the command automatically. You should also ensure the network connectivity of the KUMA system and open the ports used by its components if necessary.

2. Close the Wizard by clicking **Save collector**.

The collector service is created in KUMA. Now you will install a similar service on the server intended for receiving events.

> If a wmi or wec connector was selected for collectors, you must also install the automatically created KUMA agents.

Installing a collector in a KUMA network infrastructure

A collector consists of two parts: one part is created inside the KUMA 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. Create the /opt/kaspersky/kuma/ folder.

3. Copy the "kuma" file to the /opt/kaspersky/kuma/ folder. The file is located in the installer in the /kuma-ansible-installer/roles/kuma/files/ folder.

   Make sure the kuma file has sufficient rights to run. If the file is not executable, make it executable:

   ```
   sudo chmod +x /opt/kaspersky/kuma/kuma
   ```

4. Place the LICENSE file from the /kuma-ansible-installer/roles/kuma/files/ directory in the /opt/kaspersky/kuma/ directory and accept the license by running the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma license
   ```

5. Create the 'kuma' user:

   ```
   sudo useradd --system kuma && usermod -s /usr/bin/false kuma
   ```

6. Make the 'kuma' user the owner of the /opt/kaspersky/kuma directory and all files inside the directory:

   ```
   sudo chown -R kuma:kuma /opt/kaspersky/kuma/
   ```

7. Execute the following command:

```
sudo /opt/kaspersky/kuma/kuma collector --core https://< FQDN of the KUMA Core server >:< port used by KUMA Core for internal communication (port 7210 is used by default) > --id <service ID copied from the KUMA web interface> --api.port < port used for communication with the installed component >
```

Example: `sudo /opt/kaspersky/kuma/kuma collector --core https://test.kuma.com:7210 --id XXXX --api.port YYYY`

If errors are detected as a result of the command execution, make sure that the settings are correct. For example, the availability of the required access level, network availability between the collector service and the Core, and the uniqueness of the selected API port. After fixing errors, continue installing the collector.

If no errors were found, and the collector status in the KUMA 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. It automatically specifies the address and port of the KUMA Core server, the identifier of the collector to be installed, and the port that the collector uses for communication.

> When deploying several KUMA services on the same host, during the installation process you must specify unique ports for each component using the `--api.port <port>` parameter. The following setting values are used by default: `--api.port 7221`.

> Before installation, ensure the network connectivity of KUMA components.

8. Run the command again by adding the `--install` key:

   ```
   sudo /opt/kaspersky/kuma/kuma collector --core https://< FQDN of the KUMA Core server >:< port used by KUMA Core 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://kuma.example.com:7210 --id XXXX --api.port YYYY --install`

9. Add KUMA collector port to firewall exclusions.

   > For the program to run correctly, ensure that the KUMA components are able to interact with other components and programs over the network via the protocols and ports specified during the installation of the KUMA components.

The collector is installed. You can use it to receive data from an event source and forward it for processing.

Validating collector installation

To verify that the collector is ready to receive events:

1. In the KUMA 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 you selected **Only errors** in the **Keep raw event** drop-down list of the **Normalizer** resource in the **Resources** section of the KUMA web interface.
4. In the **Events** section of KUMA, search for events with the following parameters:
   - `ServiceID = <ID of the collector to be checked>`
   - `Raw != ""`

   If any events are found with this search, it means that there are normalization errors and they should be investigated.

To check for normalization errors using the Grafana™ Dashboard:

1. Make sure that the Collector service is running.
2. Make sure that the event source is providing events to the KUMA.
3. Open the Metrics section and follow the KUMA Collectors link.
4. See if the Errors section of the Normalization widget displays any errors.

   If there are any errors, it means that there are normalization errors and they should be investigated.

For WEC and WMI collectors, you must ensure that unique ports are used to connect to their agents. This port is specified in the **Transport** section of Collector Installation Wizard.

Ensuring uninterrupted collector operation

An uninterrupted event stream from the event source to KUMA is important for protecting the network infrastructure. Continuity can be ensured though automatic forwarding of the event stream to a larger number of collectors:

- On the KUMA side, two or more identical collectors must be installed.
- On the event source side, you must configure control of event streams between collectors using third-party server load management tools, such as rsyslog or nginx.

With this configuration of the collectors in place, no incoming events will be lost if the collector server is unavailable for any reason.

Please keep in mind that when the event stream switches between collectors, each collector will aggregate events separately.

If the KUMA collector fails to start, and its log includes the "panic: runtime error: slice bounds out of range [8:0]" error:

1. Stop the collector.

   ```
   sudo systemctl stop kuma-collector-<collector ID>
   ```

2. Delete the DNS enrichment cache files.

   ```
   sudo rm -rf /opt/kaspersky/kuma/collector/<collector ID>/cache/enrichment/DNS-*
   ```

3. Delete the event cache files (disk buffer). Run the command only if you can afford to jettison the events in the disk buffers of the collector.

   ```
   sudo rm -rf /opt/kaspersky/kuma/collector/<collector ID>/buffers/*
   ```

4. Start the collector service.

   ```
   sudo systemctl start kuma-collector-<collector ID>
   ```

Event stream control using rsyslog

To enable rsyslog event stream control on the event source server:

1. Create two or more identical collectors that you want to use to ensure uninterrupted reception of events.

2. Install rsyslog on the event source server (see the rsyslog documentation).

3. Add rules for forwarding the event stream between collectors to the configuration file /etc/rsyslog.conf:

   ```
   *. * @@ <main collector server FQDN>: <port for incoming events>
   $ActionExecOnlyWhenPreviousIsSuspended on
   *. * @@ <backup collector server FQDN>: <port for incoming events>
   $ActionExecOnlyWhenPreviousIsSuspended off
   ```

   Example configuration file

   Example configuration file specifying one primary and two backup collectors. The collectors are configured to receive events on TCP port 5140.

   ```
   *.* @@kuma-collector-01.example.com:5140
   $ActionExecOnlyWhenPreviousIsSuspended on
   & @@kuma-collector-02.example.com:5140
   & @@kuma-collector-03.example.com:5140
   $ActionExecOnlyWhenPreviousIsSuspended off
   ```

4. Restart rsyslog by running the following command:

   ```
   systemctl restart rsyslog.
   ```

   Event stream control is now enabled on the event source server.

Event stream control using nginx

To control event stream using nginx, you need to create and configure an ngnix server to receive events from the event source and then forward these to collectors.

To enable nginx event stream control on the event source server:

1. Create two or more identical collectors that you want to use to ensure uninterrupted reception of events.
2. Install nginx on the server intended for event stream control.
   - Installation command in Oracle Linux 8.6:

     ```
     $sudo dnf install nginx
     ```

   - Installation command in Ubuntu 20.4:

     ```
     $sudo apt-get install nginx
     ```

     > When installing from sources, you must compile with the parameter `-with-stream` option:
     > `$ sudo ./configure -with-stream -without-http_rewrite_module -without-http_gzip_module`
3. On the nginx server, add the stream module to the nginx.conf configuration file that contains the rules for forwarding the stream of events between collectors.

   Example stream module

Example module in which event stream is distributed between the collectors kuma-collector-01.example.com and kuma-collector-02.example.com, which receive events via TCP on port 5140 and via UDP on port 5141. Balancing uses the nginx.example.com ngnix server.

```
stream {
 upstream syslog_tcp {
server kuma-collector-1.example.com:5140;
server kuma-collector-2.example.com:5140;
}
upstream syslog_udp {
server kuma-collector-1.example.com:5141;
server kuma-collector-2.example.com:5141;
}
 server {
listen nginx.example.com:5140;
proxy_pass syslog_tcp;
}
server {
listen nginx.example.com:5141 udp;
proxy_pass syslog_udp;
proxy_responses 0;
}
}
 worker_rlimit_nofile 1000000;
events {
worker_connections 20000;
}
# worker_rlimit_nofile is the limit on the number of open files (RLIMIT_NOFILE) for workers. This is used to raise the limit without restarting the main process.
# worker_connections is the maximum number of connections that a worker can open simultaneously.
```

4. Restart nginx by running the following command:

   `systemctl restart nginx`

5. On the event source server, forward events to the ngnix server.

Event stream control is now enabled on the event source server.

Nginx Plus may be required to fine-tune balancing, but certain balancing methods, such as Round Robin and Least Connections, are available in the base version of ngnix.

For more details on configuring nginx, please refer to the nginx documentation.

Predefined collectors

The predefined collectors listed in the table below are included in the 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 forward messages**

A KUMA agent for Windows assets can be created automatically when you create a collector with the wmi or wec transport type. Although the set of resources and service of these agents are created in the Collector Installation Wizard, they must still be installed to the asset that will be used to forward a message.

Creating a set of resources for an agent

In the KUMA web interface, an agent service is created based on the set of resources for an agent that unites connectors and destinations.

To create a set of resources for an agent in the KUMA web interface:

1. In the KUMA web interface, under **Resources** → **Agents**, click **Add agent**.

   This opens a window for creating an agent with the **Base settings** tab active.

2. Specify the settings on the **Base settings** tab:

   - In the **Agent name** field, enter a unique name for the created service. The name must contain 1 to 128 Unicode characters.
   - In the **Tenant** drop-down list, select the tenant that will own the storage.

- If necessary, move the **Debug** toggle switch to the active position to enable logging of service operations.
- You can optionally add up to 256 Unicode characters describing the service in the **Description** field.

3. Click + to create a connection for the agent and switch to the added **Connection <number>** tab.

   You can remove tabs by clicking ×.

4. In the **Connector** group of settings, add a connector:

   - If you want to select an existing connector, select it from the drop-down list.
   - If you want to create a new connector, select **Create new** in the drop-down list and specify the following settings:
     - Specify the connector name in the **Name** field. The name must contain 1 to 128 Unicode characters.
     - In the **Type** drop-down list, select the connector type and specify its settings on the **Basic settings** and **Advanced settings** tabs. The available settings depend on the selected type of connector:
       - tcp
       - udp
       - nats-jetstream
       - kafka
       - http
       - file
       - ftp
       - nfs
       - wmi
       - wec
       - snmp

       The agent type is determined by the connector that is used in the agent. The only exception is for agents with a destination of the diode type. These agents are considered to be diode agents.

       When using the **tcp** or **udp** connector type at the normalization stage, IP addresses of the assets from which the events were received will be written in the DeviceAddress event field if it is empty.

       The ability to edit previously created wec or wmi connections in agents, collectors, and connectors is limited. You can change the connection type from **wec** to **wmi** and vice versa, but you cannot change the **wec** or **wmi** connection to any other connection type. At the same time, when editing other connection types, you cannot select the **wec** or **wmi** types. You can create connections without any restrictions on the types of connectors.

- You can optionally add up to 4,000 Unicode characters describing the resource in the **Description** field.

The connector is added to the selected connection of the agent's set of resources. The created connector is only available in this resource set and is not displayed in the web interface **Resources → Connectors section**.

5. In the **Destinations** group of settings, add a destination.

   - If you want to select an existing destination, select it from the drop-down list.

   - If you want to create a new destination, select **Create new** in the drop-down list and specify the following settings:

     - Specify the destination name in the **Name** field. The name must contain 1 to 128 Unicode characters.

     - In the **Type** drop-down list, select the destination type and specify its settings on the **Basic settings** and **Advanced settings** tabs. The available settings depend on the selected type of destination:

       - **nats-jetstream**—used for NATS communications.

       - **tcp**—used for communications over TCP.

       - **http**—used for HTTP communications.

       - **diode**—used to transmit events using a data diode.

       - **kafka**—used for Kafka communications.

       - **file**—used for writing to a file.

   - You can optionally add up to 4,000 Unicode characters describing the resource in the **Description** field.

   > The advanced settings for an agent destination (such as TLS mode and compression) must match the advanced destination settings for the collector that you want to link to the agent.

   There can be more than one destination point. You can add them by clicking the **Add destination** button and can remove them by clicking the ✕ button.

6. Repeat steps 3–5 for each agent connection that you want to create.

7. Click **Save**.

The set of resources for the agent is created and displayed under **Resources → Agents**. Now you can create an agent service in KUMA.

Creating an agent service in the KUMA web interface

When a set of resources is created for an agent, you can proceed to create an agent service in KUMA.

To create an agent service in the KUMA 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 set of resources 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.

Installing an agent in a KUMA network infrastructure

When an agent service is created in KUMA, you can proceed to installation of the agent to the network infrastructure assets that will be used to forward data to a collector.

> Prior to installation, verify the network connectivity of the system and open the ports used by its components.

Installing a KUMA agent on Linux assets

KUMA agent installed on Linux devices stops when you close the terminal or restart the server. To avoid starting the agents manually, we recommend installing the agent by using a system that automatically starts applications when the server is restarted, such as Supervisor. To start the agents automatically, define the automatic startup and automatic restart settings in the configuration file. For more details on configuring settings, please refer to the official documentation of automatic application startup systems. An example of configuring settings in Supervisor, which you can adapt to your needs:

```
[program:agent_<agent name>] command=sudo /opt/kaspersky/kuma/kuma agent --core
https://<KUMA Core server FQDN>:<port used by KUMA Core

autostart=true

autorestart=true
```

To install a KUMA agent to a Linux asset:

1. Log in to the server where you want to install the service.
2. Create the following directories:
   - /opt/kaspersky/kuma/
   - /opt/kaspersky/agent/
3. Copy the "kuma" file to the /opt/kaspersky/kuma/ folder. The file is located in the installer in the /kuma-ansible-installer/roles/kuma/files/ folder.

   > Make sure the kuma file has sufficient rights to run.

4. Create the 'kuma' user:

   ```
   sudo useradd --system kuma && usermod -s /usr/bin/false kuma
   ```

5. 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/
   ```

6. Execute the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma agent --core https://<KUMA Core server FQDN>:<port used by KUMA Core server for internal communication (port 7210 by default)> --id <service ID copied from the KUMA web interface> --wd <path to the directory that will contain the files of the installed agent. If this flag is not specified, the files will be stored in the directory where the kuma file is located>
   ```

   Example: `sudo /opt/kaspersky/kuma/kuma agent --core https://kuma.example.com:7210 --id XXXX --wd /opt/kaspersky/kuma/agent/XXXX`

The KUMA agent is installed on the Linux asset. The agent forwards data to KUMA, and you can set up a collector to receive this data.

Installing a KUMA agent on Windows assets

> 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.
> If you want to run the agent under a local account, you need administrator rights and Log on as a service. If you want to perform the collection remotely and only read logs under a domain account, EventLogReaders rights are sufficient.

To install 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/ folder.

2. Start the Command Prompt on the Windows asset with Administrator privileges and locate the folder containing the kuma.exe file.

3. Execute the following command:

   ```
   kuma agent --core https://< fully qualified domain name of the 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
   ```

   Example:

   ```
   kuma agent --core https://kuma.example.com:7210 --id XXXXX --user domain\username --install
   ```

   > You can get help information by executing the `kuma help agent` command.

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 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://<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>
```

Automatically created agents

When creating a collector with wec or wmi connectors, agents are automatically created for receiving Windows events.

Automatically created agents have the following special conditions:

- Automatically created agents can have only one connection.
- Automatically created agents are displayed under **Resources** → **Agents**, and `auto created` is indicated at the end of their name. Agents can be reviewed or deleted.
- The settings of automatically created agents are defined automatically based on the collector settings from the **Connect event sources** and **Transport** sections. You can change the settings only for a collector that has a created agent.
- The description of an automatically created agent is taken from the collector description in the **Connect event sources** section.
- Debugging of an automatically created agent is enabled and disabled in the **Connect event sources** section of the collector.
- When deleting a collector with an automatically created agent, you will be prompted to choose whether to delete the collector together with the agent or to just delete the collector. When deleting only the collector, the agent will become available for editing.

- When deleting automatically created agents, the type of collector changes to **http**, and the connection address is deleted from the **URL** field of the collector.

- If at least one Windows log name in wec or wmi connector is specified incorrectly, the agent will not receive events from any Windows log listed in the connector. At the same time the agent status will be green. Attempts to receive events will be repeated every 60 seconds, and error messages will be added to the service log.

In the KUMA interface, automatically created agents appear at the same time when the collector is created. However, they must still be installed on the asset that will be used to forward a message.

Update agents

When updating KUMA versions, the WMI and WEC agents installed on remote machines must also be updated.

To update the agent, use an administrator account and follow these steps:

1. In the KUMA 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://`<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

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 set of resources for a KUMA agent.

   At this step, you must create a set of resources for the KUMA agent that will receive events in an isolated network segment and prepare them for transferring to the data diode. The diode agent resource set has the following requirements:

   - The destination in the agent must have the **diode** type. In this resource, you must specify the directory from which the data diode will move files to the external network segment.

   - You cannot select connectors of the **sql** or **netflow** types for the diode agent.

   - TLS mode must be disabled in the connector of the diode agent.

3. Downloading the agent configuration file as JSON file.

   a. The set of agent resources from a diode-type destination must be downloaded as a JSON file.

   b. If secret resources were used in the agent resource set, you must manually add the secret data to the configuration file.

4. Installing the KUMA agent service in the isolated network segment.

   At this step, you must install the agent in an isolated network segment based on the agent configuration file that was created at the previous step. It can be installed to Linux and Windows devices.

Configuring a data diode

The data diode must be configured as follows:

- Data must be transferred atomically from the directory of the isolated server (where the KUMA agent places the data) to the directory of the external server (where the KUMA collector reads the data).
- The transferred files must be deleted from the isolated server.

For information on configuring the data diode, please refer to the documentation for the data diode used in your organization.

Special considerations

When working with isolated network segments, operations with SQL and NetFlow are not supported.

When using the scenario described above, the agent cannot be administered through the KUMA 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:

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 to the encrypted pfx key. The changeit value is exported from KUMA instead of the actual password. In the configuration file, you must manually specify the contents of secrets>"
}
},
"tenantID": "<ID of the tenant>"
}
```

Description of secret fields

Secret fields

Field name	Type	Description
`user`	string	User name
`password`	string	Password
`token`	string	Token
`urls`	array of strings	URL list
`publicKey`	string	Public key (used in PKI)
`privateKey`	string	Private key (used in PKI)
`pfx`	string containing the base64-encoded pfx file	Base64-encoded contents of the PFX file. In Linux, you can get the base64 encoding of a file by running the following command: `base64 -w0 src > dst`
`pfxPassword`	string	Password of the PFX
`securityLevel`	string	Used in snmp3. Possible values: `NoAuthNoPriv`, `AuthNoPriv`, `AuthPriv`
`community`	string	Used in snmp1
`authProtocol`	string	Used in snmp3. Possible values: `MD5`, `SHA`, `SHA224`, `SHA256`, `SHA384`, `SHA512`
`privacyProtocol`	string	Used in snmp3. Possible values: `DES`, `AES`
`privacyPassword`	string	Used in snmp3
`certificate`	string containing the base64-encoded pem file	Base64-encoded contents of the PEM file. In Linux, you can get the base64 encoding of a file by running the following command: `base64 -w0 src > dst`

Installing Linux Agent in an isolated network segment

To install a KUMA agent to a Linux device in an isolated network segment:

1. Place the following files on the Linux server in an isolated network segment that will be used by the agent to receive events and from which the data diode will move files to the external network segment:

   - Agent configuration file.

     > You must use an access control list (ACL) to configure access permissions for the configuration file so that only the KUMA user will have file read access.

   - Executive file /opt/kaspersky/kuma/kuma (the "kuma" file can located in the installer in the /kuma-ansible-installer/roles/kuma/files/ folder).

2. Execute the following command:

```
sudo ./kuma agent --cfg <path to the agent configuration file> --wd <path to the
directory where the files of the agent being installed will reside. If this flag is not
specified, the files will be stored in the directory where the kuma file is located>
```

The agent service is installed and running on the server in an isolated network segment. It receives events and relays them to the data diode so that they can be sent to an external network segment.

Installing Windows Agent in an isolated network segment

> Prior to installing a KUMA agent to a Windows asset, the server administrator must create a user account with the EventLogReaders and Log on as a service permissions on the Windows asset. This user account must be used to start the agent.

To install a KUMA agent to a Windows device in an isolated network segment:

1. Place the following files on the Window server in an isolated network segment that will be used by the agent to receive events and from which the data diode will move files to the external network segment:

   - Agent configuration file.

     > You must use an access control list (ACL) to configure access permissions for the configuration file so that the file can only be read by the user account that will run the agent.

   - Kuma.exe executable file. This file can be found inside the installer in the /kuma-ansible-installer/roles/kuma/files/ directory.

   It is recommended to use the `C:\Users\<user name>\Desktop\KUMA` folder.

2. Start the Command Prompt on the Windows asset with Administrator privileges and locate the folder containing the kuma.exe file.

3. Execute the following command:

   ```
   kuma.exe agent --cfg <path to the agent configuration file> --user <user name that
   will run the agent, including the domain> --install
   ```

   > You can get installer Help information by running the following command:

   ```
   kuma.exe help agent
   ```

4. Enter the password of the user account used to run the agent.

The `C:\Program Files\Kaspersky Lab\KUMA\agent\<Agent ID>` folder is created in which the KUMA agent service is installed. The agent moves events to the folder so that they can be processed by the data diode.

When installing the agent, the agent configuration file is moved to the directory C:\Program Files\Kaspersky Lab\KUMA\agent\<agent ID specified in the configuration file>. The kuma.exe file is moved to the C:\Program Files\Kaspersky Lab\KUMA directory.

When installing an agent, its configuration file must not be located in the directory where the agent is installed.

When the agent service is installed, it starts automatically. The service is also configured to restart in case of any failures.

Removing a KUMA agent from Windows assets

To remove a KUMA agent from a Windows asset:

1. Start the Command Prompt on the Windows machine with Administrator privileges and locate the folder with kuma.exe file.
2. Run any of the commands below:
   - `kuma.exe agent --cfg <path to agent configuration file> --uninstall`
   - `kuma.exe agent --id <ID of agent service that was created in KUMA> --uninstall`

   The specified KUMA agent is removed from the Windows asset. Windows events are no longer sent to KUMA.

When configuring services, you can check the configuration for errors before installation by running the agent with the following command:

```
kuma.exe agent --cfg <path to agent configuration file>
```

Transferring events from Windows machines to KUMA

To transfer events from Windows machines to KUMA, a combination of a KUMA agent and a KUMA collector is used. Data transfer is organized as follows:

1. The KUMA agent installed on the machine receives Windows events:
   - Using the WEC connector: the agent receives events arriving at the host under a subscription, as well as the server logs.
   - Using the WMI connector: the agent connects to remote servers specified in the configuration and receives events.
2. The agent sends events (without preprocessing) to the KUMA collector specified in the destination.

You can configure the agent so that different logs are sent to different collectors.

3. The collector receives events from the agent, performs a full event processing cycle, and sends the processed events to the destination.

Receiving events from the WEC agent is recommended when using centralized gathering of events from Windows hosts using Windows Event Forwarding (WEF). The agent must be installed on the server that collects events; it acts as the Windows Event Collector (WEC). We do not recommend installing KUMA agents on every endpoint host from which you want to receive events.

The process of configuring the receipt of events using the WEC Agent is described in detail in the appendix: Configuring receipt of events from Windows devices using KUMA Agent (WEC).

For details about the Windows Event Forwarding technology, please refer to the official Microsoft documentation.

We recommend receiving events using the WMI agent in the following cases:

- If it is not possible to use the WEF technology to implement centralized gathering of events, and at the same time, installation of third-party software (for example, the KUMA agent) on the event source server is prohibited.
- If you need to obtain events from a small number of hosts — no more than 500 hosts per one KUMA agent.

For connecting Windows logs as an event source, we recommend using the "Add event source" wizard . When using a wizard to create a collector with WEC or WMI connectors, agents are automatically created for receiving Windows events. You can also manually create the resources necessary for collecting Windows events.

An agent and a collector for receiving Windows events are created and installed in several stages:

1. **Creating a set of resources for an agent.**

   Agent connector:

   When creating an agent, on the **Connection** tab, you must create or select a connector of the WEC or WMI type.

   If at least one Windows log name in a WEC or WMI connector is specified incorrectly, the agent will receive events from all Windows logs listed in the connector, except the problematic log. At the same time the agent status will be green. Attempts to receive events will be repeated every 60 seconds, and error messages will be added to the service log.

   Agent destination:

   The type of agent destination depends on the data transfer method you use: nats-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 machine from which you want to receive Windows events.**

   Before installation, make sure that the system components have access to the network and open the necessary network ports:

   - Port 7210, TCP: from server with collectors to the Core.
   - Port 7210, TCP: from agent server to the Core.

- The port configured in the **URL** field when the connector was created: from the agent server to the server with the collector.

4. **Creating and installing KUMA collector.**

   When creating a set of collectors, at the **Transport** step, you must create or select a connector that the collector will use to receive events from the agent. Connector type must match the type of the agent destination.

   The advanced settings of the connector (such as delimiter, compression, and TLS mode) must match the advanced settings of the agent destination that you want to link to the agent.

Configuring event sources

This section provides information on configuring the receipt of events from various sources.

Configuring receipt of Auditd events

KUMA lets you monitor and audit the Auditd events on Linux devices.

Before configuring event receiving, make sure to create a new KUMA collector for the Auditd events.

Configuring the receipt of Auditd events involves the following steps:

1. Installation of KUMA collector in the network infrastructure.
2. Configuring the event source server.
3. Verifying receipt of Auditd events by the KUMA collector.

   You can verify that the Auditd event source server is configured correctly by searching for related events in the KUMA web interface.

Installing KUMA collector for receiving Auditd events

After creating a collector, in order to configure event receiving using rsyslog, you must install a collector on the network infrastructure server intended for receiving events.

For details on installing the KUMA collector, refer to the Installing collector in the network infrastructure section.

Configuring the event source server

The rsyslog service is used to transmit events from the server to the KUMA collector.

To configure transmission of events from the server to the collector:

1. Make sure that the rsyslog service is installed on the event source server. For this purpose, execute the following command:

```
systemctl status rsyslog.service
```

If the rsyslog service is not installed on the server, install it by executing the following command:

```
yum install rsyslog
systemctl enable rsyslog.service
systemctl start rsyslog.service
```

2. In the /etc/rsyslog.d folder , create the audit.conf file with the following content:

```
$ModLoad imfile
$InputFileName /var/log/audit/audit.log
$InputFileTag tag_audit_log:
$InputFileStateFile audit_log
$InputFileSeverity info
$InputFileFacility local6
$InputRunFileMonitor
*.* @<KUMA collector IP address>:<KUMA collector port>
```

> If you want to send events over TCP, instead of the last line in the file insert the following line: `*.* @@<KUMA collector IP address>:<KUMA collector port>`.

3. Save the changes to the audit.conf file.

4. Restart the rsyslog service by executing the following command:

```
systemctl restart rsyslog.service
```

The event source server is configured. Data about events is transmitted from the server to the KUMA collector.

Configuring receipt of KATA/EDR events

You can configure the receipt of Kaspersky Anti Targeted Attack Platform events in the KUMA SIEM system.

Before configuring event receipt, make sure to create a KUMA collector for the KATA/EDR events.

> When creating a collector in the KUMA 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.

Configuring the receipt of KATA/EDR events involves the following steps:

1. Configuring the forwarding of KATA/EDR events
2. Installing the KUMA collector in the network infrastructure

3. Verifying receipt of KATA/EDR events in the KUMA collector

   You can verify that the KATA/EDR event source server is configured correctly by searching for related events in the KUMA 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.

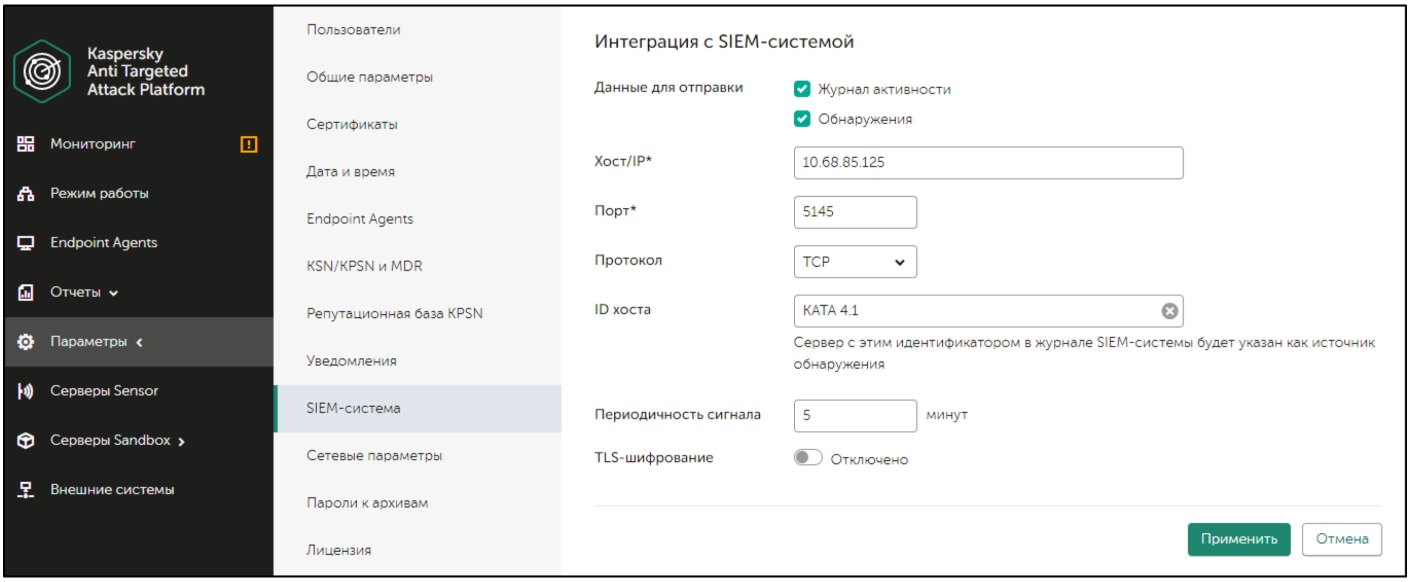
   e. In the **Host ID** field, specify the server host ID to be indicated in the SIEM systems log as a detection source.

   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.



Интеграция с SIEM-системой

Данные для отправки ☑ Журнал активности
☑ Обнаружения

Хост/IP* 10.68.85.125

Порт* 5145

Протокол TCP

ID хоста KATA 4.1

Сервер с этим идентификатором в журнале SIEM-системы будет указан как источник обнаружения

Периодичность сигнала 5 минут

TLS-шифрование ○ Отключено

Configuring Kaspersky Anti Targeted Attack Platform integration with KUMA

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 Kaspersky Security Center event receiving in CEF format

KUMA allows you to receive and export events in the CEF format from Kaspersky Security Center Administration Server to the KUMA SIEM system.

Configuring the receipt of Kaspersky Security Center events in the CEF format involves the following steps:

1. Configuring the forwarding of Kaspersky Security Center events.

2. Configuring the KUMA Collector.

3. Installing the KUMA collector in the network infrastructure.

4. Verifying receipt of Kaspersky Security Center events in the CEF format in the KUMA collector

   You can verify if the events from Kaspersky Security Center Administration Server in the CEF format 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 CEF format 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.

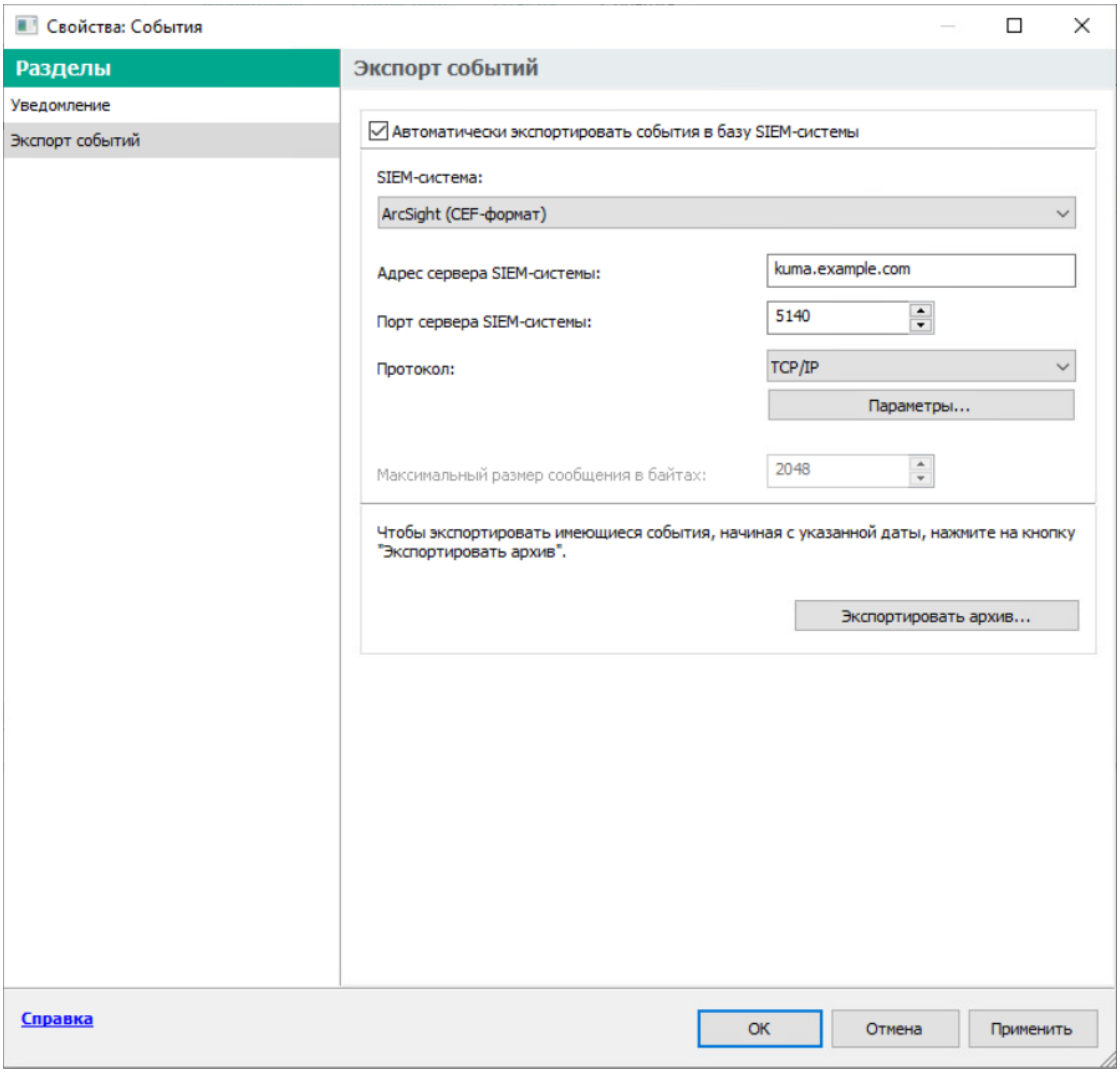
5. In the **SIEM system** drop-down list select **ArcSight (CEF format)**.

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.



Свойства: События

Экспорт событий

☑ Автоматически экспортировать события в базу SIEM-системы

SIEM-система: ArcSight (CEF-формат)

Адрес сервера SIEM-системы: kuma.example.com

Порт сервера SIEM-системы: 5140

Протокол: TCP/IP

Максимальный размер сообщения в байтах: 2048

Чтобы экспортировать имеющиеся события, начиная с указанной даты, нажмите на кнопку "Экспортировать архив".

Configuring export of Kaspersky Security Center 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. Installation of 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 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.

Configuring audit of events from Windows devices

You can configure event audit on Windows devices for an individual device or for all devices in a domain.

This section describes how to configure an audit on an individual device and how to use a domain group policy to configure an audit.

Configuring an audit policy on a Windows device

To configure audit policies on a device:

1. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.
2. In the opened window, type `secpol.msc` and click **OK**.

   The **Local security policy** window opens.
3. Select **Security Settings** → **Local policies** → **Audit policy**.
4. In the pane on the right, double-click to open the properties of the policy for which you want to enable an audit of successful and unsuccessful attempts.
5. In the **<Policy name> properties** window, on the **Local security setting** tab, select the **Success** and **Failure** check boxes to track successful and interrupted attempts.

   It is recommended to enable an audit of successful and unsuccessful attempts for the following policies:

   - Audit Logon

- Audit Policy Change
- Audit System Events
- Audit Logon Events
- Audit Account Management

Configuration of an audit policy on the device is complete.

Configuring an audit using a group policy

In addition to configuring an audit policy on an individual device, you can also configure an audit by using a domain group policy.

To configure an audit using a group policy:

1. Open the **Run** window by pressing the key combination `Win+R`.
2. In the opened window, type `gpedit.msc` and click **OK**.

   The **Local Group Policy Editor** window opens.
3. Select **Computer configuration** → **Windows configuration** → **Security settings** → **Local policies** → **Audit policy**.
4. In the pane on the right, double-click to open the properties of the policy for which you want to enable an audit of successful and unsuccessful attempts.
5. In the **<Policy name> properties** window, on the **Local security setting** tab, select the **Success** and **Failure** check boxes to track successful and interrupted attempts.

   It is recommended to enable an audit of successful and unsuccessful attempts for the following policies:

   - Audit Logon
   - Audit Policy Change
   - Audit System Events
   - Audit Logon Events
   - Audit Account Management

> If you want to receive Windows logs from a large number of servers or if installation of KUMA agents on domain controllers is not allowed, it is recommended to configure Windows log redirection to individual servers that have the Windows Event Collector service configured.

The audit policy is now configured on the server or workstation.

Configuring centralized receipt of events from Windows devices using the Windows Event Collector service

The Windows Event Collector service allows you to centrally receive data about events on servers and workstations running Windows. You can use the Windows Event Collector service to subscribe to events that are registered on remote devices.

You can configure the following types of event subscriptions:

- **Source-initiated subscriptions**. Remote devices send event data to the Windows Event Collector server whose address is specified in the group policy. For details on the subscription configuration procedure, please refer to the Configuring data transfer from the event source server section.
- **Collector-initiated subscriptions**. The Windows Event Collector server connects to remote devices and independently gathers events from local logs. For details on the subscription configuration procedure, please refer to the Configuring the Windows Event Collector service section.

Configuring data transfer from the event source server

You can receive information about events on servers and workstations by configuring data transfer from remote devices to the Windows Event Collector server.

Preliminary steps

1. Verify that the Windows Remote Management service is configured on the event source server by running the following command in the PowerShell console:

   ```
   winrm get winrm/config
   ```

   If the Windows Remote Management service is not configured, initialize it by running the following command:

   ```
   winrm quickconfig
   ```

2. If the event source server is a domain controller, make the Windows logs available over the network by running the following command in PowerShell as an administrator:

   ```
   wevtutil set-log security /ca:’O:BAG:SYD:(A;;0xf0005;;;SY)(A;;0x5;;;BA)(A;;0x1;;;S-1-5-32-573)(A;;0x1;;;S-1-5-20)
   ```

   Verify access by running the following command:

   ```
   wevtutil get-log security
   ```

Configuring the firewall on the event source server

To enable the Windows Event Collector server to receive Windows log entries, inbound connection ports must be opened on the event source server.

To open ports for inbound connections:

1. On the event source server, open the **Run** window by pressing the key combination `Win`+`R`.
2. In the opened window, type `wf.msc` and click **OK**.

   The **Windows Defender Firewall with Advanced Security** window opens.

3. Go to the **Inbound Rules** section and click **New Rule** in the **Actions** pane.

   The **New Inbound Rule Wizard** opens.

4. At the **Rule type** step, select **Port**.

5. At the **Protocols and ports** step, select **TCP** as the protocol. In the **Specific local ports** field, indicate the relevant port numbers:

   - `5985` (for HTTP access)

   - `5986` (for HTTPS access)

   You can indicate one of the ports, or both.

6. At the **Action** step, select **Allow connection** (selected by default).

7. At the **Profile** step, clear the **Private** and **Public** check boxes.

8. At the **Name** step, specify a name for the new inbound connection rule and click **Done**.

Configuration of data transfer from the event source server is complete.

> The Windows Event Collector server must have the permissions to read Windows logs on the event source server. These permissions can be assigned to both the Windows Event Collector server account and to a special user account. For details on granting permissions, please refer to the Granting user permissions to view the Windows Event Log.

Configuring the Windows Event Collector service

The Windows Event Collector server can independently connect to devices and gather data on events of any severity.

To configure the receipt of event data by the Windows Event Collector server:

1. On the event source server, open the **Run** window by pressing **`Win`**+**`R`**.

2. In the opened window, type `services.msc` and click **OK**.

   The **Services** window opens.

3. In the list of services, find and start the **Windows Event Collector** service.

4. Open the **Event Viewer** snap-in by doing the following:

   a. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.

   b. In the opened window, type `eventvwr` and click **OK**.

5. Go to the **Subscriptions** section and click **Create Subscription** in the **Actions** pane.

6. In the opened **Subscription Properties** window, specify the name and description of the subscription, and define the following settings:

   a. In the **Destination log** field, select **Forwarded events** from the list.

   b. In the **Subscription type and source computers** section, click the **Select computers** button.

c. In the opened **Computers** window, click the **Add domain computer** button.

   The **Select computer** window opens.

d. In the **Enter the object names to select (examples)** field, list the names of the devices from which you want to receive event information. Click **OK**.

e. In the **Computers** window, check the list of devices from which the Windows Event Collector server will gather event data and click **OK**.

f. In the **Subscription properties** window, in the **Collected events** field, click the **Select events** button.

g. In the opened **Request filter** window, specify how often and which data about events on devices you want to receive.

h. If necessary, in the **<All event codes>** field, list the codes of the events whose information you want to receive or do not want to receive. Click **OK**.

7. If you want to use a special account to view event data, do the following:

   a. In the **Subscription properties** window, click the **Advanced** button.

   b. In the opened **Advanced subscription settings** window, in the user account settings, select **Specific user**.

   c. Click the **User and password** button and enter the account credentials of the selected user.

Configuration of the Event Collector Service is complete.

To verify that the configuration is correct and event data is being received by the Windows Event Collector server:

In the **Event Viewer** snap-in, go to **Event Viewer (Local)** → **Windows logs** → **Forwarded events**.

Granting permissions to view Windows events

You can grant permissions to view Windows events for a specific device or for all devices in a domain.

To grant permissions to view events on a specific device:

1. Open the **Run** window by pressing the key combination `Win`+`R`.

2. In the opened window, type `compmgmt.msc` and click **OK**.

   The **Computer Management** window opens.

3. Go to **Computer Management (local)** → **Local users and groups** → **Groups**.

4. In the pane on the right, select the **Event Log Readers** group and double-click to open the policy properties.

5. Click the **Add** button at the bottom of the **Properties: Event Log Readers** window.

   The **Select Users, Computers or Groups** window opens.

6. In the **Enter the object names to select (examples)** field, list the names of the users or devices to which you want to grant permissions to view event data. Click **OK**.

To grant permissions to view events for all devices in a domain:

1. Log in to the domain controller with administrator privileges.

2. Open the **Run** window by pressing the key combination `Win`+`R`.

3. In the opened window, type `dsa.msc` and click **OK**.

   The **Active Directory Users and Computers** window opens.

4. Go to **Active Directory Users and Computers** → **<Domain name>** → **Builtin**.

5. In the pane on the right, select the **Event Log Readers** group and double-click to open the policy properties.

   In the **Properties: Event Log Readers** window, open the **Members** tab and click the **Add** button.

   The **Select Users, Computers or Groups** window opens.

6. In the **Enter the object names to select (examples)** field, list the names of the users or devices to which you want to grant permissions to view event data. Click **OK**.

Granting permissions to log on as a service

You can grant permission to log on as a service to a specific device or to all devices in a domain. The "Log on as a service" permission allows you to start a process using an account that has been granted this permission.

To grant the "Log on as a service" permission to a device:

1. Open the **Run** window by pressing the key combination `Win`+`R`.

2. In the opened window, type `secpol.msc` and click **OK**.

   The **Local security policy** window opens.

3. Go to **Security settings** → **Local policies** → **User rights assignment**.

4. In the pane on the right, double-click to open the properties of the **Log on as a service** policy.

5. In the opened **Properties: Log on as a Service** window, click the **Add User or Group** button.

   The **Select Users or Groups** window opens.

6. In the **Enter the object names to select (examples)** field, list the names of the accounts or devices to which you want to grant the permission to log on as a service. Click **OK**.

> Before granting the permission, make sure that the accounts or devices to which you want to grant the **Log on as a service** permission are not listed in the properties of the **Deny log on as a service** policy.

To grant the "Log on as a service" permission to devices in a domain:

1. Open the **Run** window by pressing the key combination `Win`+`R`.

2. In the opened window, type `gpedit.msc` and click **OK**.

   The **Local Group Policy Editor** window opens.

3. Select **Computer configuration** → **Windows configuration** → **Security settings** → **Local policies** → **User rights assignment**.

4. In the pane on the right, double-click to open the properties of the **Log on as a service** policy.

5. In the opened **Properties: Log on as a Service** window, click the **Add User or Group** button.

   The **Select Users or Groups** window opens.

6. In the **Enter the object names to select (examples)** field, list the names of the users or devices to which you want to grant the permission to log on as a service. Click **OK**.

Before granting the permission, make sure that the accounts or devices to which you want to grant the **Log on as a service** permission are not listed in the properties of the **Deny log on as a service** policy.

Configuring the KUMA Collector for receiving events from Windows devices

After you finish configuring the audit policy on devices, creating subscriptions to events and granting all the necessary permissions, you need to create a collector in the KUMA 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. In the opened **Basic event parsing** window, in the **Normalizer** field, select **[OOTB] Windows Extended v.1.0** and click **OK**.

5. At the **Routing** step, add the following destinations:

   - **Storage**. To send processed events to the storage.

   - **Correlator**. To send processed events to the correlator.

   If the **Storage** and **Correlator** destinations were not added, create them.

6. At the **Setup validation** tab, click **Create and save service**.

7. Copy the command for installing the KUMA collector that appears.

Installing the KUMA Collector for receiving events from Windows devices

After configuring the collector for receiving Windows events, install the KUMA Collector on the server of the network infrastructure intended for receiving events.

For details on installing the KUMA collector, refer to the Installing collector in the network infrastructure section.

Configuring forwarding of events from Windows devices to KUMA using KUMA Agent (WEC)

To complete the data forwarding configuration, you must create a WEC KUMA agent and then install it on the device from which you want to receive event information.

For more details on creating and installing a WEC KUMA Agent on Windows devices, please refer to the Forwarding events from Windows devices to KUMA section.

Configuring receipt of events from Windows devices using KUMA Agent (WMI)

KUMA allows you to receive information about events from Windows devices using the WMI KUMA Agent.

Configuring event receiving consists of the following steps:

1. Configuring audit settings for managing KUMA.
2. Configuring data transfer from the event source server.
3. Granting permissions to view events.
4. Granting permissions to log on as a service.
5. Creating a KUMA collector.

   To receive events from Windows devices, in the KUMA Collector Installation Wizard, at the **Event parsing** step, in the **Normalizer** field, select **[OOTB] Windows Extended v.1.0**.
6. Installing KUMA collector.
7. Forwarding events from Windows devices to KUMA.

   To complete the data forwarding configuration, you must create a WMI KUMA agent and then install it on the device from which you want to receive event information.

Configuring audit settings for managing KUMA

You can configure event audit on Windows devices both on a specific device using a local policy or on all devices in a domain using a group policy.

This section describes how to configure an audit on an individual device and how to use a domain group policy to configure an audit.

Configuring an audit using a local policy

To configure an audit using a local policy:

1. Open the **Run** window by pressing the key combination `Win`+`R`.
2. In the opened window, type `secpol.msc` and click **OK**.

   The **Local security policy** window opens.
3. Select **Security Settings** → **Local policies** → **Audit policy**.
4. In the pane on the right, double-click to open the properties of the policy for which you want to enable an audit of successful and unsuccessful attempts.
5. In the **<Policy name> properties** window, on the **Local security setting** tab, select the **Success** and **Failure** check boxes to track successful and interrupted attempts.

   It is recommended to enable an audit of successful and unsuccessful attempts for the following policies:

   - Audit Logon
   - Audit Policy Change
   - Audit System Events
   - Audit Logon Events
   - Audit Account Management

Configuration of an audit policy on the device is complete.

Configuring an audit using a group policy

In addition to configuring an audit on an individual device, you can also configure an audit by using a domain group policy.

To configure an audit using a group policy:

1. Open the **Run** window by pressing the key combination `Win`+`R`.
2. In the opened window, type `gpedit.msc` and click **OK**.

   The **Local Group Policy Editor** window opens.
3. Select **Computer configuration** → **Windows configuration** → **Security settings** → **Local policies** → **Audit policy**.
4. In the pane on the right, double-click to open the properties of the policy for which you want to enable an audit of successful and unsuccessful attempts.

5. In the **<Policy name> properties** window, on the **Local security setting** tab, select the **Success** and **Failure** check boxes to track successful and interrupted attempts.

   It is recommended to enable an audit of successful and unsuccessful attempts for the following policies:

   - Audit Logon
   - Audit Policy Change
   - Audit System Events
   - Audit Logon Events
   - Audit Account Management

The audit policy is now configured on the server or workstation.

Configuring data transfer from the event source server

Preliminary steps

1. On the event source server, open the **Run** window by pressing the key combination **`Win`**+**`R`**.
2. In the opened window, type `services.msc` and click **OK**.

   The **Services** window opens.
3. In the list of services, find the following services:
   - Remote Procedure Call
   - RPC Endpoint Mapper
4. Check the **Status** column to confirm that these services have the **Running** status.

Configuring the firewall on the event source server

The Windows Management Instrumentation server can receive Windows log entries if ports are open for inbound connections on the event source server.

To open ports for inbound connections:

1. On the event source server, open the **Run** window by pressing the key combination **`Win`**+**`R`**.
2. In the opened window, type `wf.msc` and click **OK**.

   The **Windows Defender Firewall with Advanced Security** window opens.
3. In the **Windows Defender Firewall with Advanced Security** window, go to the **Inbound Rules** section and in the **Actions** pane, click **New Rule**.

   This opens the **New Inbound Rule Wizard**.

4. In the **New Inbound Rule Wizard,** at the **Rule Type** step, select **Port**.
5. At the **Protocols and ports** step, select **TCP** as the protocol. In the **Specific local ports** field, indicate the relevant port numbers:
   - `135`
   - `445`
   - `49152–65535`
6. At the **Action** step, select **Allow connection** (selected by default).
7. At the **Profile** step, clear the **Private** and **Public** check boxes.
8. At the **Name** step, specify a name for the new inbound connection rule and click **Done**.

Configuration of data transfer from the event source server is complete.

Granting permissions to view Windows events

You can grant permissions to view Windows events for a specific device or for all devices in a domain.

To grant permissions to view events on a specific device:

1. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.
2. In the opened window, type `compmgmt.msc` and click **OK**.

   The **Computer Management** window opens.
3. Go to **Computer Management (local)** → **Local users and groups** → **Groups**.
4. In the pane on the right, select the **Event Log Readers** group and double-click to open the policy properties.
5. Click the **Add** button at the bottom of the **Properties: Event Log Readers** window.

   The **Select Users, Computers or Groups** window opens.
6. In the **Enter the object names to select (examples)** field, list the names of the users or devices to which you want to grant permissions to view event data. Click **OK**.

To grant permissions to view events for all devices in a domain:

1. Log in to the domain controller with administrator privileges.
2. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.
3. In the opened window, type `dsa.msc` and click **OK**.

   The **Active Directory Users and Computers** window opens.
4. In the **Active Directory Users and Computers** window, go to the **Active Directory Users and Computers** section → **<Domain name>** → **Builtin**.
5. In the pane on the right, select the **Event Log Readers** group and double-click to open the policy properties.

In the **Properties: Event Log Readers** window, open the **Members** tab and click the **Add** button.

The **Select Users, Computers or Groups** window opens.

6. In the **Select User, Computer, or Group** window, In the **Enter the object name to select (examples)** field, list the names of the users or devices to which you want to grant permissions to view event data. Click **OK**.

Granting permissions to log on as a service

You can grant permission to log on as a service to a specific device or to all devices in a domain. The "Log on as a service" permission allows you to start a process using an account that has been granted this permission.

Before granting the permission, make sure that the accounts or devices to which you want to grant the **Log on as a service** permission are not listed in the properties of the **Deny log on as a service** policy.

To grant the "Log on as a service" permission to a device:

1. Open the **Run** window by pressing the key combination `Win+R`.
2. In the opened window, type `secpol.msc` and click **OK**.

   The **Local security policy** window opens.
3. In the **Local Security Policy** window, go to the **Security Settings** → **Local Policies** → **User Rights Assignment** section.
4. In the pane on the right, double-click to open the properties of the **Log on as a service** policy.
5. This opens the **Properties: Log on as a Service** window; in that window, click **Add User or Group**.

   This opens the **Select Users or Groups** window.
6. In the **Enter the object names to select (examples)** field, list the names of the accounts or devices to which you want to grant the permission to log on as a service. Click **OK**.

To grant the "Log on as a service" permission to devices in a domain:

1. Open the **Run** window by pressing the key combination `Win+R`.
2. In the opened window, type `gpedit.msc` and click **OK**.

   The **Local Group Policy Editor** window opens.
3. Select **Computer configuration** → **Windows configuration** → **Security settings** → **Local policies** → **User rights assignment**.
4. In the pane on the right, double-click to open the properties of the **Log on as a service** policy.
5. This opens the **Properties: Log on as a Service** window; in that window, click **Add User or Group**.

   This opens the **Select Users or Groups** window.
6. In the **Enter the object names to select (examples)** field, list the names of the users or devices to which you want to grant the permission to log on as a service. Click **OK**.

Configuring receipt of PostgreSQL events

KUMA lets you monitor and audit PostgreSQL events on Linux devices using rsyslog.

Events are audited using the pgAudit plugin. The plugin supports PostgreSQL 9.5 and later. For details about the pgAudit plugin, see https://github.com/pgaudit/pgaudit.

Configuring event receiving consists of the following steps:

1. Installing the pdAudit plugin.
2. Creating a KUMA collector for PostgreSQL events.

   To receive PostgreSQL events using rsyslog, in the collector installation wizard, at the **Event parsing** step, select the **[OOTB] PostgreSQL pgAudit syslog** normalizer.
3. Installing a collector in the KUMA network infrastructure.
4. Configuring the event source server.
5. Verifying receipt of PostgreSQL events in the KUMA collector

   You can verify that the PostgreSQL event source server is correctly configured in the Searching for related events section of the KUMA 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 that has administrative privileges.
2. In the **System message forwarding** menu, move the **Syslog** toggle switch to the **enabled** position.
3. For the **IP address** setting, specify the IP address of the KUMA collector.
4. For the Port setting, enter the port that the KUMA collector is listening on.
5. Click **Save** to apply the changes.

The forwarding of Ideco UTM events to KUMA is configured.

Configuring receipt of KWTS events

You can configure the receipt of events from the Kaspersky Web Traffic Security (KWTS) web traffic analysis and filtering system in KUMA.

Configuring event receiving consists of the following steps:

1. Configuring export of KWTS events to KUMA.
2. Creating a KUMA collector for receiving KWTS events.

   To receive KWTS events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] KWTS** normalizer.
3. Installing a KUMA collector for receiving KWTS events.
4. Verifying receipt of KWTS events in the KUMA collector.

   You can verify that KWTS event export is correctly configured in the Searching for related events section of the KUMA 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. Configuring export of KLMS events to KUMA
2. Creating a KUMA collector for receiving KLMS events

   To receive KLMS events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] KLMS syslog CEF** normalizer.

3. Installing a KUMA collector for receiving KLMS events
4. Verifying receipt of KLMS events in the KUMA collector

   You can verify that the KLMS event source server is correctly configured in the Searching for related events section of the KUMA 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) 1.1 mail traffic analysis and filtering system in the KUMA SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring export of KSMG events to KUMA

2. Creating a KUMA collector for receiving KSMG events

   To receive KSMG events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] KSMG** normalizer.

3. Installing a KUMA collector for receiving KSMG events

4. Verifying receipt of KSMG events in the KUMA collector

   You can verify that the KSMG event source server is correctly configured in the Searching for related events section of the KUMA web interface.

Configuring export of KSMG events to KUMA

To configure the export of KSMG events to KUMA:

1. Connect to the KSMG server via SSH using an account with administrator rights.

2. Use the ksmg-control utility to download the settings to the settings.xml file:

   ```
   sudo /opt/kaspersky/ksmg/bin/ksmg-control --get-settings EventLogger -n -f
   /tmp/settings.xml
   ```

3. Make sure that the settings in the /tmp/settings.xml file have the following values; make changes if necessary:

   ```
   <siemSettings>
   <enabled>1</enabled>
   <facility>Local1</facility>
   ```

4. Apply settings with the following command:

   ```
   sudo /opt/kaspersky/ksmg/bin/ksmg-control --set-settings EventLogger -n -f
   /tmp/settings.xml
   ```

5. To send events via UDP, make the following changes to the /etc/rsyslog.conf configuration file:

   ```
   $WorkDirectory /var/lib/rsyslog
   $ActionQueueFileName ForwardToSIEM
   $ActionQueueMaxDiskSpace 1g
   $ActionQueueSaveOnShutdown on
   $ActionQueueType LinkedList
   $ActionResumeRetryCount -1
   local1.* @<<IP address of the KUMA collector>:<port of the collector>>
   ```

   If you want to send events over TCP, the last line should be as follows:

   ```
   local1.* @@<<IP address of the KUMA collector>:<port of the collector>>
   ```

6. Save your changes.

7. Restart the rsyslog service with the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

Configuring receipt of PT NAD events

You can configure the receipt of PT NAD events in the KUMA SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring export of PT NAD events to KUMA.

2. Creating a KUMA collector for receiving PT NAD events.

   To receive PT NAD events using Syslog, in the Collector Installation Wizard, at the **Event parsing** step, select the [OOTB] PT NAD json normalizer.

3. Installing a KUMA collector for receiving PT NAD events.

4. Verifying receipt of PT NAD events in the KUMA collector.

   You can verify that the PT NAD event source server is correctly configured in the Searching for related events section of the KUMA 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 a sample configuration of sending syslog messages with information about activities, attacks, and indicators of compromise to two remote servers via TCP and UDP without writing to the local log:

```
notifier.yaml.syslog_notifier.addresses:
remote1:
address: tcp://198.51.100.1:1514
remote2:
address: udp://198.51.100.2:2514
```

4. Save your changes in the /opt/ptsecurity/etc/ptdpi.settings.yaml.

5. Restart the ptdpi-worker@notifier module:

   ```
   sudo ptdpictl restart-worker notifier
   ```

The sending of events to KUMA via Syslog is configured.

Configuring receipt of events using the MariaDB Audit Plugin

KUMA allows auditing events using the MariaDB Audit Plugin. The plugin supports MySQL 5.7 and MariaDB. The audit plugin does not support MySQL 8. Detailed information about the plugin is available on the official MariaDB website.

We recommend using MariaDB Audit Plugin version 1.2 or later.

Configuring event receiving consists of the following steps:

1. Configuring the MariaDB Audit Plugin to send MySQL events and configuring the Syslog server to send events.

2. Configuring the MariaDB Audit Plugin to send MariaDB events and configuring the Syslog server to send events.

3. Creating a KUMA Collector for MySQL 5.7 and MariaDB Events.

   To receive MySQL 5.7 and MariaDB events using the MariaDB Audit Plugin, in the KUMA Collector Installation Wizard, at the **Event parsing** step, in the **Normalizer** field, select **[OOTB] MariaDB Audit Plugin syslog**.

4. Installing a collector in the KUMA network infrastructure.

5. Verifying receipt of MySQL and MariaDB events by the KUMA collector.

   To verify that the MySQL and MariaDB event source server is configured correctly, you can search for related events.

Configuring the MariaDB Audit Plugin to send MySQL events

The MariaDB Audit Plugin is supported for MySQL 5.7 versions up to 5.7.30 and is bundled with MariaDB.

To configure MySQL 5.7 event reporting using the MariaDB Audit Plugin:

1. Download the MariaDB distribution kit and extract it.

   You can download the MariaDB distribution kit from the official MariaDB website. The operating system of the MariaDB distribution must be the same as the operating system on which MySQL 5.7 is running.

2. Connect to MySQL 5.7 using an account with administrator rights by running the following command:

   ```
   mysql -u <username> -p
   ```

3. To get the directory where the MySQL 5.7 plugins are located, on the MySQL 5.7 command line, run the following command:

   ```
   SHOW GLOBAL VARIABLES LIKE 'plugin_dir'
   ```

4. In the directory obtained at step 3, copy the MariaDB Audit Plugin from `<directory to which the distribution kit was extracted>`/mariadb-server-`<version>`/lib/plugins/server_audit.so.

5. On the operating system command line, run the following command:

   ```
   chmod 755 <directory to which the distribution kit was extracted> server_audit.so
   ```

   For example:

   ```
   chmod 755 /usr/lib64/mysql/plugin/server_audit.so
   ```

6. On the MySQL 5.7 command line, run the following command:

   ```
   install plugin server_audit soname 'server_audit.so'
   ```

7. Create a backup copy of the /etc/mysql/mysql.conf.d/mysqld.cnf configuration file.

8. In the configuration file /etc/mysql/mysql.conf.d/mysqld.cnf, in the `[mysqld]` section, add the following lines:

   ```
   server_audit_logging=1
   server_audit_events=connect,table,query_ddl,query_dml,query_dcl
   server_audit_output_type=SYSLOG
   server_audit_syslog_facility=LOG_SYSLOG
   ```

   If you want to disable event export for certain audit event groups, remove some of the values from the `server_audit_events` setting. Descriptions of settings are available on the MariaDB Audit Plugin vendor's website.

9. Save changes to the configuration file.

10. Restart the MariaDB service by running one of the following commands:

    - `systemctl restart mysqld` for a system with systemd initialization.
    - `service mysqld restart` for a system with init initialization.

MariaDB Audit Plugin for MySQL 5.7 is configured. If necessary, you can run the following commands on the MySQL 5.7 command line:

- `show plugins` to check the list of current plugins.
- `SHOW GLOBAL VARIABLES LIKE 'server_audit%'` to check the current audit settings.

Configuring the MariaDB Audit Plugin to send MariaDB Events

The MariaDB Audit Plugin is included in the MariaDB distribution kit starting with versions 5.5.37 and 10.0.10.

To configure MariaDB event export using the MariaDB Audit Plugin:

1. Connect to MariaDB using an account with administrator rights by running the following command:

   ```
   mysql -u <username> -p
   ```

2. To check if the plugin is present in the directory where operating system plugins are located, run the following command on the MariaDB command line:

   ```
   SHOW GLOBAL VARIABLES LIKE 'plugin_dir'
   ```

3. On the operating system command line, run the following command:

   ```
   ll <directory obtained by the previous command> | grep server_audit.so
   ```

   If the command output is empty and the plugin is not present in the directory, you can either copy the MariaDB Audit Plugin to that directory or use a newer version of MariaDB.

4. On the MariaDB command line, run the following command:

   ```
   install plugin server_audit soname 'server_audit.so'
   ```

5. Create a backup copy of the /etc/mysql/my.cnf configuration file.

6. In the /etc/mysql/my.cnf configuration file, in the `[mysqld]` section, add the following lines:

   ```
   server_audit_logging=1
   server_audit_events=connect,table,query_ddl,query_dml,query_dcl
   server_audit_output_type=SYSLOG
   server_audit_syslog_facility=LOG_SYSLOG
   ```

   If you want to disable event export for certain audit event groups, remove some of the values from the `server_audit_events` setting. Descriptions of settings are available on the MariaDB Audit Plugin vendor's website.

7. Save changes to the configuration file.

8. Restart the MariaDB service by running one of the following commands:

   - `systemctl restart mariadb` for a system with systemd initialization.
   - `service mariadb restart` for a system with init initialization.

MariaDB Audit Plugin for MariaDB is configured. If necessary, you can run the following commands on the MariaDB command line:

- `show plugins` to check the list of current plugins.

- `SHOW GLOBAL VARIABLES LIKE 'server_audit%'` to check the current audit settings.

Configuring a Syslog server to send events

The rsyslog service is used to transmit events from the server to the collector.

To configure the sending of events from the server where MySQL or MariaDB is installed to the collector:

1. Before making any changes, create a backup copy of the /etc/rsyslog.conf configuration file.
2. To send events via UDP, add the following line to the /etc/rsyslog.conf configuration file:

   ```
   *.* @<IP address of the KUMA collector>:<port of the KUMA collector>
   ```

   For example:

   ```
   *.* @192.168.1.5:1514
   ```

   If you want to send events over TCP, the line should be as follows:

   ```
   *.* @@192.168.1.5:2514
   ```

   Save changes to the /etc/rsyslog.conf configuration file.
3. Restart the rsyslog service by executing the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

Configuring receipt of Apache Cassandra events

KUMA allows receiving information about Apache Cassandra events.

Configuring event receiving consists of the following steps:

1. Configuring Apache Cassandra event logging in KUMA.
2. Creating a KUMA collector for Apache Cassandra events.

   To receive Apache Cassandra events, in the KUMA Collector Installation Wizard, at the **Transport** step, select a **file** type connector; at the **Event parsing** step, in the **Normalizer** field, select **[OOTB] Apache Cassandra file**.
3. Installing a collector in the KUMA network infrastructure.
4. Verifying receipt of Apache Cassandra events in the KUMA collector.

   To verify that the Apache Cassandra event source server is configured correctly, you can search for related events.

Configuring Apache Cassandra event logging in KUMA

To configuring Apache Cassandra event logging in KUMA:

1. Make sure that the server where Apache Cassandra is installed has 5 GB of free disk space.

2. Connect to the Apache Cassandra server using an account with administrator rights.

3. Before making changes, create backup copies of the following configuration files:

   - /etc/cassandra/cassandra.yaml

   - /etc/cassandra/logback.xml

4. Make sure that the settings in the /etc/cassandra/cassandra.yaml configuration file have the following values; make changes if necessary:

   a. in the `audit_logging_options` section, set the `enabled` setting to `true`.

   b. in the `logger` section, set the `class_name` setting to `FileAuditLogger`.

5. Add the following lines to the /etc/cassandra/logback.xml configuration file:

```
<!-- Audit Logging (FileAuditLogger) rolling file appender to audit.log -->
<appender name="AUDIT" class="ch.qos.logback.core.rolling.RollingFileAppender">
<file>${cassandra.logdir}/audit/audit.log</file>
<rollingPolicy class="ch.qos.logback.core.rolling.SizeAndTimeBasedRollingPolicy">
<!-- rollover daily -->
<fileNamePattern>${cassandra.logdir}/audit/audit.log.%d{yyyy-MM-dd}.%i.zip</fileNamePattern>
<!-- each file should be at most 50MB, keep 30 days worth of history, but at most 5GB -->
<maxFileSize>50MB</maxFileSize>
<maxHistory>30</maxHistory>
<totalSizeCap>5GB</totalSizeCap>
</rollingPolicy>
<encoder>
<pattern>%-5level [%thread] %date{ISO8601} %F:%L - %replace(%msg){'\n', ' '}%n</pattern>
</encoder>
</appender>
<!-- Audit Logging additivity to redirect audt logging events to audit/audit.log -->
<logger name="org.apache.cassandra.audit" additivity="false" level="INFO">
<appender-ref ref="AUDIT"/>
</logger>
```

6. Save changes to the configuration file.

7. Restart the Apache Cassandra service using the following commands:

   a. `sudo systemctl stop cassandra.service`

   b. `sudo systemctl start cassandra.service`

8. After restarting, check the status of Apache Cassandra using the following command:

```
sudo systemctl status cassandra.service
```

Make sure that the command output contains the following sequence of characters:

```
Active: active (running)
```

Apache Cassandra event export is configured. Events are located in the /var/log/cassandra/audit/ directory, in the audit.log file (${cassandra.logdir}/audit/audit.log).

Configuring receipt of FreeIPA events

You can configure the receipt of FreeIPA events in KUMA via the Syslog protocol.

Configuring event receiving consists of the following steps:

1. Configuring export of FreeIPA events to KUMA.
2. Creating a KUMA collector for receiving FreeIPA events.

   To receive FreeIPA events, in the KUMA Collector Setup Wizard, at the **Event parsing** step, in the **Normalizer** field, select **[OOTB] FreeIPA**.
3. Installing the KUMA collector in the network infrastructure.
4. Verifying receipt of FreeIPA events by KUMA.

   To verify that the FreeIPA event source server is configured correctly, you can search for related events.

Configuring export of FreeIPA events to KUMA

To configure the export of FreeIPA events to KUMA via the Syslog protocol in JSON format:

1. Connect to the FreeIPA server via SSH using an account with administrator rights.
2. In the /etc/rsyslog.d/ directory, create a file named freeipa-to-siem.conf.
3. Add the following lines to the /etc/rsyslog.d/freeipa-to-siem.conf configuration file:

```
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")
}
*.* @<IP address of the KUMA collector>:<port of the KUMA collector KUMA>;ls_json
```

You can fill in the last line in accordance with the selected protocol:

`*.* @<192.168.1.10>:<1514>;ls_json` for sending events over UDP

`*.* @@<192.168.2.11>:<2514>;ls_json` for sending events over TCP

4. Add the following lines to the /etc/rsyslog.conf configuration file:

   ```
   $IncludeConfig /etc/freeipa-to-siem.conf
   $RepeatedMsgReduction off
   ```

5. Save changes to the configuration file.

6. Restart the rsyslog service by executing the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

Configuring receipt of VipNet TIAS events

You can configure the receipt of ViPNet TIAS events in KUMA via the Syslog protocol.

Configuring event receiving consists of the following steps:

1. Configuring export of ViPNet TIAS events to KUMA.

2. Creating a KUMA collector for receiving ViPNet TIAS events.

   To receive ViPNet TIAS events using Syslog, in the Collector Installation Wizard, at the **Event parsing** step, select the [OOTB] Syslog-CEF normalizer.

3. Installing a KUMA collector for receiving ViPNet TIAS events.

4. Verifying receipt of ViPNet TIAS events in KUMA.

   You can verify that ViPNet TIAS event source server is correctly configured in the Searching for related events section of the KUMA 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, set the **Do not send incident information in CEF format** toggle switch to enabled.

    As a result, when new incidents are detected or the statuses of previously detected incidents change, depending on the statuses selected during configuration, the corresponding information is sent to the specified addresses of receiving servers via the syslog protocol in CEF format.

8. Click **Save changes**.

Export of events to the KUMA collector is configured.

Configuring receipt of Nextcloud events

You can configure the receipt of Nextcloud 26.0.4 events in 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 administrative privileges.

2. Edit the Snort configuration file. To do so, run the following command on the command line:

   ```
   sudo vi /usr/local/etc/snort/snort.lua
   ```

3. In the configuration file, edit the alert_json block:

   ```
   alert_json =
   {
   ```

```
file = true,
limit = 500,
fields = 'seconds action class b64_data dir dst_addr dst_ap dst_port eth_dst eth_len \
eth_src eth_type gid icmp_code icmp_id icmp_seq icmp_type iface ip_id ip_len msg mpls
\
pkt_gen pkt_len pkt_num priority proto rev rule service sid src_addr src_ap src_port \
target tcp_ack tcp_flags tcp_len tcp_seq tcp_win tos ttl udp_len vlan timestamp',
}
```

4. To complete the configuration, run the following command:

   ```
   sudo /usr/local/bin/snort -c /usr/local/etc/snort/snort.lua -s 65535 -k none -l
   /var/log/snort -i <name of the interface that Snort is listening on> -m 0x1b
   ```

As a result, Snort events are logged to /var/log/snort/alert_json.txt.

Configuring receipt of Suricata events

You can configure the receipt of Suricata 7.0.1 events in 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 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 administrative privileges.
2. Create a backup copy of the FreeRADIUS configuration file:

   ```
   sudo cp /etc/freeradius/3.0/radiusd.conf /etc/freeradius /3.0/radiusd.conf.bak
   ```
3. Open the FreeRADIUS configuration file for editing:

   ```
   sudo nano /etc/freeradius/3.0/radiusd.conf
   ```
4. In the 'log' section, edit the settings as follows:

```
destination = syslog
syslog_facility = daemon
stripped_names = no
auth = yes
auth_badpass = yes
auth_goodpass = yes
```

5. Save the configuration file.

FreeRADIUS event audit is configured.

Configuring a Syslog server to send FreeRADIUS events

The rsyslog service is used to transmit events from the FreeRADIUS server to the KUMA collector.

To configure the sending of events from the server where FreeRADIUS is installed to the collector:

1. In the /etc/rsyslog.d/ directory, create the FreeRADIUS-to-siem.conf file and add the following line to it:

   ```
   If $programname contains 'radiusd' then @<IP address of the collector>:<port of the collector>
   ```

   If you want to send events via TCP, the contents of the file must be as follows:

   ```
   If $programname contains 'radiusd' then @<IP address of the collector>:<port of the collector>
   ```

2. Create a backup copy of the /etc/rsyslog.conf file.

3. Add the following lines to the /etc/rsyslog.conf configuration file:

   ```
   $IncludeConfig /etc/FreeRADIUS-to-siem.conf
   $RepeatedMsgReduction off
   ```

4. Save your changes.

5. Restart the rsyslog service:

   ```
   sudo systemctl restart rsyslog.service
   ```

The export of events from the FreeRADIUS server to the KUMA collector is configured.

Configuring receipt of VMware vCenter events

You can configure the receipt of VMware vCenter events in the KUMA SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring the connection to VMware vCenter.

2. Creating a KUMA collector for receiving VMware vCenter events.

   To receive VMware vCenter events, in the collector installation wizard, at the **Transport** step, select the vmware connector type. Specify the required settings:

- The URL at which the VMware API is available, for example, https://vmware-server.com:6440.
- VMware credentials — a secret that specifies the username and password for connecting to the VMware API.

At the **Event parsing** step, select the [OOTB] VMware vCenter API normalizer.

3. Installing a KUMA collector for receiving VMware vCenter events.
4. Verifying receipt of VMware vCenter events in the KUMA collector.

   You can verify that the VMware vCenter event source server is correctly configured in the Searching for related events section of the KUMA 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 that has administrative privileges.
2. Go to the Security&Users section and select Users.
3. Create a user account.
4. Go to the Roles section and assign the "Read-only: See details of objects role, but not make changes" role to the created account.

   You will use the credentials of this user account in the secret of the collector.

   For details about creating user accounts, refer to the VMware vCenter documentation.

The connection to VMware vCenter for receiving events is configured.

Configuring receipt of zVirt events

You can configure the receipt of zVirt 3.1 events in 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 machine that is running the HostedEngine virtual machine and click **Edit**.
3. In the **Edit virtual machine** window, go to the **Logging** section.
4. Select the **Determine Syslog server address** check box.
5. In the text box, enter the collector information in the following format: `<IP address or FQDN of the KUMA collector>: <port of the KUMA collector>`.
6. If you want to use TCP instead of UDP for sending logs, select the **Use TCP connection** check box.

Event export is configured.

Configuring receipt of Zeek IDS events

You can configure the receipt of Zeek IDS 1.8 events in 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.

   To receive Zeek IDS events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] ZEEK IDS json** file normalizer, and at the **Transport** step, select the **file** connector type.

3. Installing KUMA collector for receiving Zeek IDS events

4. Verifying receipt of Zeek IDS events in the KUMA collector

   You can verify that the Zeek IDS event source server is correctly configured in the Searching for related events section of the KUMA web interface.

New Topic (202)

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 administrative privileges.
2. Create the directory where JSON event logs must be stored:
   ```
   sudo mkdir /opt/zeek/logs/zeek-json
   ```
3. Change to this directory:
   ```
   sudo cd /opt/zeek/logs/zeek-json
   ```
4. Run the command that uses the jq utility to convert the original event log format to the target format:
   ```
   jq . -c <path to the log file to be converted to a different format> >> <new file
   name>.log
   ```
   Example:
   ```
   jq . -c /opt/zeek/logs/current/conn.log >> conn.log
   ```
   As a result of running the command, a new file is created in the /opt/zeek/logs/zeek-json directory if this file did not exist before. If the file was already present in the current directory, new information is appended to the end of the file.

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. KUMA creates event sources based on the following fields of events (the data in these fields is case sensitive):

- DeviceProduct is a required field.
- One of the DeviceHostname or DeviceAddress fields must be present.
- DeviceProcessName is an optional field.
- Tenant is a required field, which is determined automatically from the tenant of the event that was used to identify the source.

Limitations

1. KUMA registers an event source, provided that the DeviceAddress and DeviceProduct fields are contained in a raw event.

   If the raw event does not contain the DeviceAddress and DeviceProduct fields, you can do the following:

- Configure enrichment in the normalizer: select the **Event** data type on the **Enrichment** tab of the normalizer, specify the **Source field** setting, select DeviceAddress and DeviceProduct as the **Target field**, and click OK.

- Use an enrichment rule: select the **Event** data source type, specify the **Source field** setting, select DeviceAddress and DeviceProduct as the **Target field**, and 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 DeviceProduct + DeviceHostname + DeviceAddress required fields, KUMA registers different sources if the following conditions are satisfied:

- The values of the required fields are identical, but different tenants are determined for the events.

- The values of the required fields are identical, but one of the events has an optional DeviceProcessName field specified.

- The values of the required fields are identical, but the data in these fields have different character case.

If you want KUMA to log such events under the same source, you can further configure the fields in the normalizer.

Lists of sources are generated in collectors, merged in the KUMA Core, and displayed in the program web interface under **Source status** on the List of event sources tab. Data is updated every minute.

The rate and number of incoming events serve as an important indicator of the state of the observed system. You can configure monitoring policies such that changes are tracked automatically and notifications are automatically created when indicators reach specific boundary values. Monitoring policies are displayed in the KUMA 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

Sources of events are displayed in the table under **Source status** → **List of event sources**. One page can display up to 250 sources. You can sort the table by clicking the column header of the relevant setting. Clicking on a source of events opens an incoming data graph.

You can use the **Search** field to search for event sources. The search is performed using regular expressions (RE2).

If necessary, you can configure the interval for updating data in the table. Available update periods: **1 minute**, **5 minutes**, **15 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.

The following columns are available:

- **Status**—status of the event source:
  - Green—events are being received within the limits of the assigned monitoring policy.
  - Red—the frequency or number of incoming events go beyond the boundaries defined in the monitoring policy.

  - Gray—a monitoring policy has not been assigned to the source of events.

  The table can be filtered by this setting.

- **Name**—name of the event source. The name is generated automatically from the following fields of events:
  - DeviceProduct
  - DeviceAddress and/or DeviceHostName
  - DeviceProcessName
  - Tenant

  You can change the name of an event source. The name can contain no more than 128 Unicode characters.

- **Host name or IP address**—host name or IP address from which the events were forwarded.
- **Monitoring policy**—name of the monitoring policy assigned to the event source.
- **Stream**—frequency at which events are received from the event source.
- **Lower limit**—lower boundary of the permissible number of incoming events as indicated in the monitoring policy.
- **Upper limit**—upper boundary of the permissible number of incoming events as indicated in the monitoring policy.
- **Tenant**—the tenant that owns the events received from the event source.

If you select sources of events, the following buttons become available:

- **Save to CSV**—you can use this button to export data of the selected event sources to a file named event-source-list.csv in UTF-8 encoding.
- **Apply policy** and **Disable policy**—you can use these buttons to enable or disable a monitoring policy for a source of events. When enabling a policy, you must select the policy from the drop-down list. When disabling a policy, you must select how long you want to disable the policy: temporarily or forever.

  If there is no policy for the selected event source, the **Apply policy** button is inactive. This button will also be inactive if sources from different tenants are selected, but the user has no available policies in the shared tenant.

  > In some rare cases, the status of a disabled policy may change from gray to green a few seconds after it is disabled due to overlapping internal processes of KUMA. If this happens, you need to disable the monitoring policy again.

- **Remove event source from the list**—you can use this button to remove an event source 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.

> By default, no more than 250 event sources are displayed and, therefore, available for selection. If there are more event sources, to select them you must load additional event sources by clicking the **Show next 250** button in the lower part of the window.

Monitoring policies

The rate and number of incoming events serve as an important indicator of the state of the system. For example, you can detect when there are too many events, too few, or none at all. Monitoring policies are designed to detect such situations. In a policy, you can specify a lower threshold, an optional upper threshold, and the way the events are counted: by frequency or by total number.

The policy must be applied to the event source. After applying the policy, you can monitor the status of the source: green means everything is OK, red means the stream is outside the configured threshold. If the status is red, an event of the Monitoring type generated. You can also configure notifications to be sent to an arbitrary email address. 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. Clicking a policy opens the data area with policy settings. The settings can be edited.

To add a monitoring policy:

1. In the KUMA web interface, under **Source status** → **Monitoring policies**, click **Add policy** and define the settings in the opened window:

   a. In the **Policy name** field, enter a unique name for the policy 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 policy. Your tenant selection determines the specific sources of events that can covered by the monitoring policy.

   c. In the **Policy type** drop-down list, select one of the following options:

      - **byCount**—by the number of events over a certain period of time.

      - **byEPS**—by the number of events per second over a certain period of time. The average value over the entire period is calculated. You can additionally track spikes during specific periods.

   d. 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.

   e. In the **Count interval** field, specify the period during which the monitoring policy must take into account the data from the monitoring source. The maximum value is 14 days.

   f. If necessary, specify the email addresses to which notifications about the activation of the KUMA monitoring policy should be sent. To add each address, click the **Email** button.

      To forward notifications, you must configure a connection to the SMTP server.

2. Click **Add**.

The monitoring policy will be added.

To remove a monitoring policy,

select one or more policies, then click **Delete policy** and confirm the action.

> You cannot remove preinstalled monitoring policies or policies that have been assigned to data sources.

Managing assets

Assets represent the computers of the organization. You can add assets to KUMA; in that case, KUMA automatically adds asset IDs when enriching events, and when you analyze events, you can get additional information about computers in the organization.

You can add assets to KUMA in the following ways:

- Import assets:
  - From the MaxPatrol report.
  - On a schedule from Kaspersky Security Center and KICS for Networks.

    By default, assets are imported every 12 hours, this frequency can be configured. On-demand import of assets is also possible; such on-demand import does not affect the scheduled import time. From the Kaspersky Security Center database, 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 information about hte computer: name, address, time of connection to Kaspersky Security Center, information about hardware and software, including the operating system, and information about vulnerabilities. That is, the information that is collected by means of the Kaspersky Security Center Network Agent.
- Create assets manually through the web interface or via the API.

  You can add assets manually. In this case, you must manually specify the following information: address, FQDN, name and version of the operating system, hardware information. Information about the vulnerabilities of assets cannot be added through the web interface. You can provide information about vulnerabilities if you add assets using the API.

You can manage KUMA assets: view information about assets, search for assets, add, edit or delete assets, and export asset data to a CSV file.

Asset categories

You can categorize the assets and then use the categories in filter conditions or correlation rules. For example, you can create alerts of a higher severity level for assets from a higher-severity category. By default, all assets fall into the **Uncategorized assets** category. A device can be added to multiple categories.

By default, KUMA assigns the following severity levels to asset categories: Low, Medium, High, Critical. You can create custom categories, categories can be nested.

Categories can be populated in the following ways:

- **Manually**
- **Active**: dynamic if the asset **meets the specified conditions**. For example, the moment the asset is upgraded to a specified OS version or placed in a specified subnet, the asset is moved to the specified category.

1. In the **Repeat categorization every** drop-down list, specify how often assets will be linked to a category. You can select values ranging from once per hour to once per 24 hours.

   You can forcibly start categorization by selecting **Start categorization** in the category context menu.

2. In the **Conditions** settings block, specify the filter for matching assets to attach to an asset category.

   You can add conditions by clicking the **Add condition** buttons. Groups of conditions can be added by using the **Add group** buttons. Group operators can be switched between **AND**, **OR**, and **NOT** values.

   Categorization filter operands and operators

Operand	Operators	Comment
Build number	>, >=, =, <=, <	
OS	=, like	The "like" operator ensures that the search is not case sensitive.
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	=, like	The "like" operator ensures that the search is not case sensitive.
CVE	=, in	The "in" operator lets you specify an array of values.
Software	=, like	
CII	in	More than one value can be selected.
Anti-virus databases last updated	>=,<=	
Last update of the information	>=,<=	
Protection last updated	>=,<=	
System last 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	>=,<=	

3. Use the **Test conditions** button to make sure that the specified filter is correct. When you click the button, you should see the **Assets for given conditions** window containing a list of assets that satisfy the search conditions.

- **Reactive**—When a correlation rule is triggered, the asset is moved to the specified group.

In 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 binding of assets to the selected category. This option is available for categories that have active categorization.
Add subcategory	Add a subcategory to the selected category.
Edit category	Edit the selected category.
Delete category	Delete the selected category. You can only delete categories that have no assets or subcategories. Otherwise the **Delete category** option is inactive.
Pin as tab	Display the selected category on a separate tab. You can undo this action by selecting **Unpin as tab** in the context menu of the relevant category.

Adding an asset category

To add an asset category:

1. Open the **Assets** section in the KUMA web interface.

2. Open the category creation window:

   - Click the **Add category** button.

- If you want to create a subcategory, select **Add subcategory** in the context menu of the parent category.

The **Add category** details area appears in the right 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.

   - In the **Parent** field, indicate the position of the category within the categories tree hierarchy:

     a. Click the button.

        This opens the **Select categories** window showing the categories tree. If you are creating a new category and not a subcategory, the window may show multiple asset category trees, one for each tenant that you can access. Your tenant selection in this window cannot be undone.

     b. Select the parent category for the category you are creating.

     c. Click **Save**.

     Selected category appears in **Parent** fields.

   - The **Tenant** field displays the tenant whose structure contains your selected parent category. The tenant category cannot be changed.

   - Assign a severity to the category in the **Priority** drop-down list.

   - If necessary, in the **Description** field, you can add a note consisting of up to 256 Unicode characters.

4. In the **Categorization kind** drop-down list, select how the category will be populated with assets. Depending on your selection, you may need to specify additional settings:

   - **Manually**—assets can only be manually linked to a category.

   - **Active**—assets will be assigned to a category at regular intervals if they satisfy the defined filter.

     Active category of assets

1. In the **Repeat categorization every** drop-down list, specify how often assets will be linked to a category. You can select values ranging from once per hour to once per 24 hours.

   You can forcibly start categorization by selecting **Start categorization** in the category context menu.

2. In the **Conditions** settings block, specify the filter for matching assets to attach to an asset category.

   You can add conditions by clicking the **Add condition** buttons. Groups of conditions can be added by using the **Add group** buttons. Group operators can be switched between **AND**, **OR**, and **NOT** values.

   Categorization filter operands and operators ?

Operand	Operators	Comment
Build number	>, >=, =, <=, <	
OS	=, like	The "like" operator ensures that the search is not case sensitive.
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	=, like	The "like" operator ensures that the search is not case sensitive.
CVE	=, in	The "in" operator lets you specify an array of values.
Software	=, like	
CII	in	More than one value can be selected.
Anti-virus databases last updated	>=,<=	
Last update of the information	>=,<=	
Protection last updated	>=,<=	
System last 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	>=,<=	

3. Use the **Test conditions** button to make sure that the specified filter is correct. When you click the button, you should see the **Assets for given conditions** window containing a list of assets that satisfy the search conditions.

- **Reactive**—the category will be filled with assets by using correlation rules.

5. Click **Save**.

The new category will be added to the asset categories tree.

Configuring the table of assets

In 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 ⚙ 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**

   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:

1. Make sure that the 🔍 button is enabled in the upper left part of the **Assets** section of the KUMA web interface.

   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

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	=, >, >=, <, <=	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.
Asset source	in	• Kaspersky Security Center • KICS for Networks • Imported via API • Created manually
RAM	=, >, >=, <, <=	Number.
Number of disks	=, >, >=, <, <=	Number.
Number of network cards	=, >, >=, <, <=	Number.
Disk free bytes	=, >, >=, <, <=	Number.
Anti-virus databases last updated	>=, <=	Date.
Last update of the information	>=, <=	Date.
Protection last updated	>=, <=	Date.
System last started	>=, <=	Date.
KSC extended status	in	• The host with the Network Agent installed is connected to the network, but the Network Agent is not active • The anti-virus application is installed, but real-time protection is not enabled • Anti-virus application is installed but not running

		• The number of detected viruses is too large • The anti-virus application is installed, but the real-time protection status differs from the one set by the security administrator • The anti-virus application is not installed • A full virus scan was performed too long ago • The anti-virus databases were updated too long ago • The Network Agent is inactive for too long • License expired • The number of untreated objects is too large • Restart required • Incompatible applications are installed on the host • Vulnerabilities are detected on the host • The last scan for operating system updates on the host was too long ago • Invalid encryption status of the host • Mobile device settings do not comply with security policy requirements • Unprocessed incidents detected • Host status is suggested by a managed product • Insufficient disk space on the host. Synchronization errors occur, or not enough disk space
Real-time protection status	=	• Suspended • Starting • Running (if the anti-virus application does not support the Running status categories) • Performed with maximum protection • Performed with maximum performance • Performed with recommended settings • Performed with custom settings • Error
Encryption	=	• Encryption rules are not configured on the host.

status		• 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 • In progress • Error • Not installed • License is missing
Anti-virus protection status of mail servers	=	• Unknown • Stopped • Suspended • Starting • In progress • Error • Not installed • License is missing
Data Leakage Prevention status	=	• Unknown • Stopped • Suspended • Starting

		• In progress • Error • Not installed • License is missing
KSC extended status ID	=	• OK • Critical • Attention required
Endpoint Sensor status	=	• Unknown • Stopped • Suspended • Starting • In progress • Error • Not installed • License is missing
Last visible	>=, <=	Date

To find an asset:

1. Make sure that the ▤ button is enabled in the upper left part of the **Assets** section of the KUMA web interface.

   The asset filtering settings are displayed in the upper part of the window.

2. Specify the asset filtering settings and click the **Search** button.

The table displays the assets that meet the search criteria.

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 for Networks, 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 for Networks vulnerabilities**—vulnerabilities of the asset, if provided. This information is available for the assets imported from KICS for Networks.

- 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 for Networks server IP address** and **KICS for Networks connector ID**—data on the KICS for Networks 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, "The anti-virus databases were updated too long ago".

  - **Real-time protection status** – status of Kaspersky applications installed on the asset. For example: "Running (if the anti-virus application does not support the Running status categories)".

  - **Encryption status** – information about asset encryption. For example: "Encryption rules are not configured on the host".

  - **Spam protection status** – status of anti-spam protection. For example, "Started".

- **Anti-virus protection status of mail servers** – status of the virus protection of mail servers. For example, "Started".

- **Data Leakage Prevention status** – status of data leak protection. For example, "Started".

- **Endpoint Sensor status** – status of data leak protection. For example, "Started".

- **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.

Clicking the **KSC response** button starts Kaspersky Security Center task on the asset and clicking the **Move to KSC group** button moves the asset being viewed between Kaspersky Security Center administration groups.

> This is available if KUMA is integrated with Kaspersky Security Center.

Adding assets

You can add asset information in the following ways:

- Manually.

  You can add an asset using the KUMA web interface or the API.

- Import assets.

  You can import assets from Kaspersky Security Center, KICS for Networks, and MaxPatrol reports.

When assets are added, assets that already exist in KUMA can be merged with the assets being added.

Asset merging algorithm:

1. Checking uniqueness of Kaspersky Security Center or KICS for Networks assets.

   - The uniqueness of an asset imported from Kaspersky Security Center is determined by the **Host ID** parameter, which contains the Kaspersky Security Center Network Agent *Network Agent* identifier. If two assets' IDs differ, they are considered to be separate assets and are not merged.

   - The uniqueness of an asset imported from KICS for Networks is determined by the combination of the **IP address**, **KICS for Networks server IP address**, and **KICS for Networks connector ID** parameters. If any of the parameters of two assets differ they are considered to be separate assets and are not merged.

   If the compared assets match, the algorithm is performed further.

2. Make sure that the values in the **IP**, **MAC**, and **FQDN** fields match.

If at least two of the specified fields match, the assets are combined, provided that the other fields are blank.

Possible matches:

- The FQDN and IP address of the assets match. The **MAC** field is blank.

  The check is performed against the entire array of IP address values. If the IP address of an asset is included in the FQDN, the values are considered to match.

- The FQDN and MAC address of the assets match. The **IP** field is blank.

  The check is performed against the entire array of MAC address values. If at least one value of the array fully matches the FQDN, the values are considered to match.

- The IP address and MAC address of the assets match. The **FQDN** field is blank.

  The check is performed against the entire array of IP- and MAC address values. If at least one value in the arrays is fully matched, the values are considered to match.

3. Make sure that the values of at least one of the **IP**, **MAC**, or **FQDN** fields match, provided that the other two fields are not filled in for one or both assets.

   Assets are merged if the values in the field match. For example, if the FQDN and IP address are specified for a KUMA asset, but only the IP address with the same value is specified for an imported asset, the fields match. In this case, the assets are merged.

   For each field, verification is performed separately and ends on the first match.

You can see examples of asset field comparison here.

Information about assets can be generated from various sources. If the added asset and the 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 for Networks.
Owner	The first value from the sources is selected according to the following priority: • Received from Kaspersky Security Center. • Manually defined.
IP address	The data is merged. If the array of addresses contains identical addresses, the copy of the duplicate address is deleted.
FQDN	The first value from the sources is selected according to the following priority:

	• Received from Kaspersky Security Center. • Received by KICS for Networks. • Manually defined.
MAC address	The data is merged. If the array of addresses contains identical addresses, one of the duplicate addresses is deleted.
Operating system	The first value from the sources is selected according to the following priority: • Received from Kaspersky Security Center. • Received by KICS for Networks. • Manually defined.
Vulnerabilities	KUMA asset data is supplemented with information from the added assets. In the asset details, data is grouped by the name of the source. Vulnerabilities are eliminated for each source separately.
Software info	Data from KICS for Networks is always recorded (if available). For other sources, the first value is selected according to the following priority: • Received from Kaspersky Security Center. • Manually defined.
Hardware info	The first value from the sources is selected according to the following priority: • Received from Kaspersky Security Center. • Defined via the API.

The updated data is displayed in the asset details. You can view asset details in 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 for Networks, you can edit the following asset data:

- Name.
- Category.

If asset information was added manually, you can edit the following asset data when editing these assets in 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.

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 ☰ button.

      Select categories window opens.

   b. Select the check boxes next to the categories that should be assigned to the asset. You can use the ⊞ and ⊟ icons to expand or collapse the lists of categories.

   c. Click **Save**.

   The selected categories appear in the **Categories** fields.

4. If required, add information about the operating system installed on the asset in the **Software** section.

5. If required, add information about asset hardware in the **Hardware info** section.

6. Click **Add**.

The asset is created and displayed in the assets table in the category assigned to it or in the **Uncategorized assets** category.

Importing asset information from Kaspersky Security Center

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

KUMA supports the following types of asset imports from KSC:

- Import of information about all assets of all KSC servers.
- Import of information about assets of the selected KSC server.

To import information about all assets of all KSC servers:

1. In the KUMA web interface, select the **Assets** section.
2. Click the **Import assets** button.

   The **Import Kaspersky Security Center assets** window opens.
3. In the drop-down list, select the tenant for which you want to perform the import.

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

   If you want to import information about all assets of all KSC servers for all tenants, select **All tenants**.
4. Click **OK**.

The asset information will be imported.

To import information about the assets of one KSC server:

1. Open the KUMA web interface and select **Settings** → **Kaspersky Security Center**.

   The **Kaspersky Security Center integration by tenant** window opens.
2. Select the tenant for which you want to import assets.

   The **Kaspersky Security Center integration** window opens.
3. Click the connection for the relevant Kaspersky Security Center server.

   This opens a window containing the settings of this connection to Kaspersky Security Center.
4. Do one of the following:
   - If you want to import all assets connected to the selected KSC server, click the **Import assets** button.
   - If you want to import only assets that are connected to a secondary server or included in one of the groups (for example, the Unassigned devices group), do the following:

     a. Click the **Load hierarchy** button.

b. Select the check boxes next to the names of the secondary servers or groups from which you want to import asset information.

c. Select the **Import assets from new groups** check box if you want to import assets from new groups.

If no check boxes are selected, information about all assets of the selected KSC server is uploaded during the import.

d. Click **Save**.

e. Click the **Import assets** button.

The asset information will be imported.

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 1.1.

  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 /users/whoami and POST /api/v1/assets/import have been configured.

  > To import assets from MaxPatrol, it is recommended to create a separate user with the minimum necessary set of rights to use API requests.

3. Copy the maxpatrol-tool to the server hosting the KUMA Core and make the tool's file executable by running the following command:

   ```
   chmod +x <path to the maxpatrol-tool file on the server hosting the KUMA Core>
   ```

4. Run the maxpatrol-tool:

   ```
   ./maxpatrol-tool --kuma-rest <KUMA REST API server address and port> --token <path and name of API token file> --tenant <name of tenant where assets will reside> <path and name of MaxPatrol report file> --cert <path to the KUMA Core certificate file>
   ```

   Example: `./maxpatrol-tool --kuma-rest example.kuma.com:7223 --token token.txt --tenant Main example.xml --cert /opt/kaspersky/kuma/core/certificates/ca.cert`

   You can use additional flags and commands for import operations. For example, the command `--verbose, -v` will display a full report on the received assets. A detailed description of the available flags and commands is provided in the table titled Flags and commands of maxpatrol-tool. You can also use the `--help` command to view information on the available flags and commands.

The asset information will be imported from the MaxPatrol report to KUMA. The console displays information on the number of new and updated assets.

```
Example:
inserted 2 assets;
updated 1 asset;
errors occurred: []
```

The tool works as follows when importing assets:

- KUMA overwrites the data of assets imported through the API, and deletes information about their resolved vulnerabilities.
- KUMA skips assets with invalid data. Error information is displayed when using the `--verbose` flag.
- If there are assets with identical IP addresses and fully qualified domain names (FQDN) in the same MaxPatrol report, these assets are merged. The information about their vulnerabilities and software is also merged into one asset.

  > When uploading assets from MaxPatrol, assets that have equivalent IP addresses and fully qualified domain names (FQDN) that were previously imported from Kaspersky Security Center are overwritten.

  To avoid this problem, you must configure range-based asset filtering by running the following command:

  ```
  --ignore <IP address ranges> or -i <IP address ranges>
  ```

  Assets that satisfy the filtering criteria are not uploaded. For a description of this command, please refer to the table titled *Flags and commands of maxpatrol-tool*.

Flags and commands of maxpatrol-tool

Flags and commands	Description
`--kuma-rest <KUMA REST API server port and address>, -a <KUMA REST API server port and address>`	Address (with the port) of KUMA Core server where assets will be imported. For example, `example.kuma.com:7223`. Port 7223 is used for API requests by default. You can change the port if necessary.
`--token <path and name of API token file>, -t <path and name of API token file>`	Path and name of the file containing the token used to access the REST API. This file must contain only the token. The 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. By default, the certificate is located in the folder with the application installed: /opt/kaspersky/kuma/core/certificates/ca.cert.

Examples:

- `./maxpatrol-tool --kuma-rest example.kuma.com:7223 --token token.txt --tenant Main example.xml --cert /example-directory/ca.cert` – import assets to KUMA from MaxPatrol report example.xml.
- `./maxpatrol-tool help`—get reference information on the tool.

Possible errors

Error message	Description

must provide path to xml file to import assets	The path to the MaxPatrol report file was not specified.
incorrect IP address format	Invalid IP address format. This error may arise when incorrect IP ranges are indicated.
no tenants match specified name	No suitable tenants were found for the specified tenant name using the REST API.
unexpected number of tenants (%v) match specified name. Tenants are: %v	KUMA returned more than one tenant for the specified tenant name.
could not parse file due to error: %w	Error reading the XML file containing the MaxPatrol report.
error decoding token: %w	Error reading the API token file.
error when importing files to KUMA: %w	Error transferring asset information to KUMA.
skipped asset with no FQDN and IP address	One of the assets in the report did not have an FQDN or IP address. Information about this asset was not sent to KUMA.
skipped asset with invalid FQDN: %v	One of the assets in the report had an incorrect FQDN. Information about this asset was not sent to KUMA.
skipped asset with invalid IP address: %v	One of the assets in the report had an incorrect IP address. Information about this asset was not sent to KUMA.
KUMA response: %v	An error occurred with the specified report when importing asset information.
unexpected status code %v	An unexpected HTTP code was received when importing asset information from KUMA.

Importing asset information from MaxPatrol VM

The 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. Importing data is supported for MaxPatrol VM 1.1.

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 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 a file with the token for accessing the KUMA REST API. 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.
   - In the user account, under **API access rights**, the check box is selected for POST/api/v1/assets/import.

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
   ```

2. Copy the kuma-ptvm-config-template.yaml template to create a configuration file named kuma-ptvm-config.yaml:

   ```
   cp kuma-ptvm-config-template.yaml kuma-ptvm-config.yaml
   ```

3. Edit the settings in the kuma-ptvm-config.yaml configuration file.

4. Save the changes to the file.

The configuration file will be created. 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:

   a. Run the kuma-ptvm utility with the following options:

   ```
   kuma-ptvm --config <path to the kuma-ptvm-config.yaml file> --download
   ```

   For each tenant specified in the configuration file, a separate file is created with a name of the form <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`	An optional setting in the 'General settings' group. Logging level.	Available values: - `trace`

		- `info` - `warning` - `error` Default setting: `info`.
`period`	An optional setting in the 'General settings' group. Data for assets that have changed during the specified period is exported from MaxPatrol.	No limitations apply. Default setting: 30d.
`strict_import`	Optional setting in the 'General settings' group. When exporting assets from MaxPatrol, check if the required fields for KUMA are filled. Do not export unverified assets from MaxPatrol.	Available values: - `true` to check for the presence of fields that are required for KUMA. - `false` to skip the check for the presence of fields that are required for KUMA. Default setting: `false`. We recommend specifying `true` when exporting assets from MaxPatrol, this lets you detect and fix possible errors in JSON files before you import assets into KUMA.
`endpoint`	Required setting in the 'KUMA settings' group. URL of the KUMA API server. For example, kuma-example.com:7223	-
`token`	Required setting in the 'KUMA settings' group. KUMA API token.	-
`ignore_server_cert`	Optional setting in the 'KUMA settings' group. 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' group. URL of the MaxPatrol API server.	-
`user`	Required setting in the 'MaxPatrol VM' group. MaxPatrol API user name.	-

`password`	Required setting in the 'MaxPatrol VM' group. MaxPatrol API user password.	-
`secret`	Required setting in the 'MaxPatrol VM settings' group. MaxPatrol API secret.	-
`ignore_server_cert`	Optional setting in the 'MaxPatrol VM settings' group. Validation of the MaxPatrol certificate.	Available values: • `true` to disable the validation of the MaxPatrol certificate. • `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' group. Export from MaxPatrol only assets with vulnerabilities for which exploits are known.	Available values: • `true` to export only assets with vulnerabilities for which exploits are known. • `false` to export all assets. Default setting: `false`.
`min_severity`	Optional setting in the 'Vulnerability filter' group. Import only vulnerabilities of the specified level or higher.	Available values: • `low` • `medium` • `high` • `critical` Default value: `low`.
`id`	Required setting in the 'Tenant map' group. Tenant ID in 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' group. Regular expression for searching the FQDN of an asset.	-
`networks`	Optional setting in the 'Tenant map' group. One or more subnets.	-
`default_tenant`	Optional setting. The default KUMA tenant for data about assets that could not be allocated to tenants specified in the 'Tenants' group of settings.	-

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 for Networks asset data for a tenant:

1. In the KUMA web interface, open **Settings** → **Kaspersky Industrial CyberSecurity for Networks**.
2. Select the relevant tenant.

   The **Kaspersky Industrial CyberSecurity for Networks integration** window opens.
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.

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 for Networks.
   - **IP address** and/or **FQDN** (required) You can specify multiple FQDNs separated by commas.
   - **MAC address**
   - **Owner**
   - **Software info**:
     - **OS name**
     - **OS build**
   - **Hardware info**:

     Hardware parameters ?

> You can add information about asset hardware to the **Hardware info** section:
>
> Available fields for describing the asset CPU:
>
> - **CPU name**
> - **CPU frequency**
> - **CPU core count**
>
>   You can add CPUs to the asset by using the **Add CPU** link.
>
> Available fields for describing the asset disk:
>
> - **Disk free bytes**
> - **Disk volume**
>
>   You can add disks to the asset by using the **Add disk** link.
>
> Available fields for describing the asset RAM:
>
> - **RAM frequency**
> - **RAM total bytes**
>
> Available fields for describing the asset network card:
>
> - **Network card name**
> - **Network card manufacture**
> - **Network card driver version**
>
>   You can add network cards to the asset by using the **Add network card** link.

- Custom fields.
- CII category.

4. Assign or change the category of the asset:

   a. Click the button.
   Select categories window opens.

   b. Select the check boxes next to the categories that should be assigned to the asset.

   c. Click **Save**.

   The selected categories appear in the **Categories** fields.

> You can also select the asset and then drag and drop it into the relevant category. This category will be added to the list of asset categories.

> Do not assign the `Categorized assets` category to assets.

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 KSC 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 **Archived assets retention period** 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 KSC 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 for Networks.

- Disabled integration with Kaspersky Security Center.

  If you disable integration with Kaspersky Security Center, the asset is considered active for 30 days. After 30 days, the asset is automatically **archived and is stored in the database for the time specified in the Archived assets retention period**.

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 dies not exist in Kaspersky Security Center or KICS for Networks.

- Connection with the Kaspersky Security Center server has not been established for more than 30 days.

To configure the archived assets retention period:

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

   This opens the **Assets** window.

2. Enter the new value in the **Archived assets retention period** 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 is available to all roles except first line analyst. 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.

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.

You can see the specific group of an asset 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 for Networks.
- 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 program monitors manual deletion of assets, as well as automatic deletion of assets imported from Kaspersky Security Center and KICS for Networks, whose data is no longer being received.
- Vulnerability info was added to the asset. The program monitors the appearance of new vulnerability data for assets. Information about vulnerabilities can be added to an asset, for example, when importing assets from Kaspersky Security Center or KICS for Networks.

- Asset vulnerability was resolved. The program monitors the removal of vulnerability information from an asset. A vulnerability is considered to be resolved if data about this vulnerability is no longer received from any sources from which information about its occurrence was previously obtained.
- Asset was added to a category. The program monitors the assignment of an asset category to an asset.
- Asset was removed from a category. The program monitors the deletion of an asset from an asset category.

By default, if asset audit is enabled, under the conditions described above, 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 settings block of the relevant type of asset audit events, use the **Add destination** drop-down list to select the type of destination to which the created events should be sent:

   - Select **Storage** if you want events to be sent to storage.
   - Select **Correlator** if you want events to be sent to the correlator.
   - Select **Other** if you want to select a different destination.

     > This type of resource includes correlator and storage services that were created in previous versions of the program.

   In the **Add destination** window that opens you must define the settings for event forwarding.

   b. Use the **Destination** drop-down list to select an existing destination or select **Create** if you want to create a new destination.

If you are creating a new destination, fill in the settings as indicated in the destination description.

c. Click **Save**.

A destination has been added to the condition for generating asset audit events. Multiple destinations can be added for each condition.

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 **Settings** → **Asset audit** and select the tenant for which you want to enable or disable an 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 **Settings** → **Asset audit** 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.

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 lines with the mouse 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 importance category**.
   - **CII object of the third importance category**.
   - **CII object of the second importance category**.
   - **CII object of the first importance category**.

3. Click **Save**.

Integration with other solutions

In this section, you'll learn how to integrate KUMA with other solutions to enrich its functionality.

Integration with Kaspersky Security Center

You can 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. Different tasks may require different access rights.

   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 a secret of the credentials type for connecting to Kaspersky Security Center**

3. **Configuring Kaspersky Security Center integration settings**

4. **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.

Configuring Kaspersky Security Center integration settings

To configure the settings for integration with Kaspersky Security Center:

1. Open the KUMA web interface and select **Settings** → **Kaspersky Security Center**.

   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. For the **Disabled** check box, do one of the following:

   - Clear the check box if you want to enable integration with Kaspersky Security Center for this tenant.

   - Select the check box if you want to disable integration with Kaspersky Security Center for this tenant.

   This check box is cleared by default.

4. In the **Data refresh interval** field, specify the time interval at which KUMA updates data on Kaspersky Security Center devices.

   The interval is specified in hours and must be an integer.

   The default time interval is 12 hours.

5. Click the **Save** button.

The Kaspersky Security Center integration settings for the selected tenant will be configured.

If the required tenant is not in the list of tenants, you need to add it to the list.

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 **Settings** → **Kaspersky Security Center**.

   The **Kaspersky Security Center integration by tenant** window opens.

2. Click the **Add tenant** button.

   The **Kaspersky Security Center integration** window opens.

3. In the **Tenant** drop-down list, select the tenant that you need to add.

4. Click the **Save** button.

The selected tenant will be added to 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 **Settings** → **Kaspersky Security Center**.

   The **Kaspersky Security Center integration by tenant** window opens.

2. Select the tenant for which you want to create a connection to Kaspersky Security Center.

3. Click the **Add connection** button and define the values for the following settings:

   - **Name** (required)—the name of the connection. The name can contain 1 to 128 Unicode characters.

   - **URL** (required)—the URL of the Kaspersky Security Center server in hostname:port or IPv4:port format.

   - In the **Secret** drop-down list, select the secret with the Kaspersky Security Center account credentials or **create a new secret** ?.

> 1. Click the + button.
>
>   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 ✎.

- **Disabled**—the state of the connection to the selected Kaspersky Security Center server. If the check box is selected, 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.

  This check box is cleared by default.

4. If you want KUMA to import only assets that are connected to secondary servers or included in groups:

   a. Click the **Load hierarchy** button.

   b. Select the check boxes next to the names of the secondary servers and groups from which you want to import asset information.

   c. If you want to import assets only from new groups, select the **Import assets from new groups** check box.

   If no check boxes are selected, information about all assets of the selected Kaspersky Security Center server is uploaded during the import.

5. 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 **Settings** → **Kaspersky Security Center**.

   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. Click the Kaspersky Security Center connection you want to change.

   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 **Settings** → **Kaspersky Security Center**.

   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. 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** → **Connectors** section, find the relevant predefined connector in the folder hierarchy, select the check box next to that connector, and click **Duplicate**.

   b. This opens the **Create connector** window; in that window, on the **Basic settings** tab, in the **Default query** field, if necessary, replace the KAV database name with the name of the Kaspersky Security Center database you are using.

      An example of a query to the Kaspersky Security Center SQL database ?

```
SELECT ev.event_id AS externalId, ev.severity AS severity, ev.task_display_name AS taskDisplayName,
    ev.product_name AS product_name, ev.product_version AS product_version,
     ev.event_type As deviceEventClassId, ev.event_type_display_name As event_subcode, ev.descr As msg,
CASE
    WHEN ev.rise_time is not NULL THEN DATEADD(hour,DATEDIFF(hour,GETUTCDATE(),GETDATE()),ev.rise_time )
      ELSE ev.rise_time
    END
  AS endTime,
  CASE
    WHEN ev.registration_time is not NULL
      THEN DATEADD(hour,DATEDIFF(hour,GETUTCDATE(),GETDATE()),ev.registration_time )
      ELSE ev.registration_time
    END
  AS kscRegistrationTime,
  cast(ev.par7 as varchar(4000)) as sourceUserName,
  hs.wstrWinName as dHost,
  hs.wstrWinDomain as strNtDom, serv.wstrWinName As kscName,
    CAST(hs.nIp / 256 / 256 / 256 % 256 AS VARCHAR) + '.' +
  CAST(hs.nIp / 256 / 256 % 256 AS VARCHAR) + '.' +
  CAST(hs.nIp / 256 % 256 AS VARCHAR) + '.' +
  CAST(hs.nIp % 256 AS VARCHAR) AS sourceAddress,
  serv.wstrWinDomain as kscNtDomain,
    CAST(serv.nIp / 256 / 256 / 256 % 256 AS VARCHAR) + '.' +
  CAST(serv.nIp / 256 / 256 % 256 AS VARCHAR) + '.' +
  CAST(serv.nIp / 256 % 256 AS VARCHAR) + '.' +
  CAST(serv.nIp % 256 AS VARCHAR) AS kscIP,
  CASE
  WHEN virus.tmVirusFoundTime is not NULL

      THEN DATEADD(hour,DATEDIFF(hour,GETUTCDATE(),GETDATE()),virus.tmVirusFoundTime )

      ELSE ev.registration_time

    END

  AS virusTime,

  virus.wstrObject As filePath,

  virus.wstrVirusName as virusName,

  virus.result_ev as result

FROM KAV.dbo.ev_event as ev

LEFT JOIN KAV.dbo.v_akpub_host as hs ON ev.nHostId = hs.nId

INNER JOIN KAV.dbo.v_akpub_host As serv ON serv.nId = 1

Left Join KAV.dbo.rpt_viract_index as Virus on ev.event_id = virus.nEventVirus

where registration_time >= DATEADD(minute, -191, GetDate())
```

c. Place the cursor in the **URL** field and in the displayed list, click ✎ in the line of the secret that you are using.

d. This opens the **Secret** window; in that window, in the **URL** field, specify the server connection address in the following format:

`sqlserver://user:password@kscdb.example.com:1433/database`

where:

- `user`—user account with public and db_datareader rights to the required database.
- `password`—user account password.
- `kscdb.example.com:1433`—address and port of the database server.
- `database`—name of the Kaspersky Security Center database. 'KAV' by default.

Click **Save**.

e. In the **Create connector** window, in the **Connection** section, in the **Query** field, replace the 'KAV' database name with the name of the Kaspersky Security Center database you are using.

You must do this if you want to use the ID column to which the query refers.

Click **Save**.

2. Install the collector in the web interface:

a. Start the Collector Installation Wizard in one of the following ways:

- In 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**.

b. At step 1 of the installation wizard, **Connect event sources**, specify the collector name and select the tenant.

c. At step 2 of the installation wizard, **Transport**, select the copy of the connector that you created at step 1.

d. At step 3 of the installation wizard, **Event parsing**, on the **Parsing schemes** tab, click **Add event parsing**.

e. This opens the **Basic event parsing** window; in that window, on the **Normalization scheme** tab, select **[OOTB] KSC from SQL** in the **Normalizer** drop-down list and click **OK**.

f. If necessary, specify the other settings in accordance with your requirements for the collector. For the purpose of importing events, editing settings at the remaining steps of the Installation Wizard is optional.

g. At step 8 of the installation wizard, **Setup validation**, click **Create and save service**.

The lower part of the window displays the command that you must use to install the collector on the server. Copy this command to the clipboard.

h. Close the Collector Installation Wizard by clicking **Save collector**.

3. Install the collector on the server.

To do so, on the server on which you want to receive Kaspersky Security Center events, run the command that you copied to the clipboard after creating the collector in the web interface.

As a result, the collector is installed and can receive events from the SQL database of Kaspersky Security Center.

You can view Kaspersky Security Center events in the **Events** section of the web interface.

Kaspersky Endpoint Detection and Response integration

Kaspersky Endpoint Detection and Response (hereinafter also referred to as "KEDR") is a functional unit of Kaspersky Anti Targeted Attack Platform that protects assets in an enterprise LAN.

You can configure KUMA integration with Kaspersky Endpoint Detection and Response 4.1 and 5.0 to manage threat response actions on assets connected to Kaspersky Endpoint Detection and Response servers, and on Kaspersky Security Center assets. Commands to perform operations are received by the Kaspersky Endpoint Detection and Response server, which then relays those commands to the Kaspersky Endpoint Agent installed on assets.

You can also import events to KUMA and receive information about Kaspersky Endpoint Detection and Response alerts (for more details, 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 program component administrator menu is displayed.
3. In the program component administrator menu, select **Technical Support Mode**.
4. Press **Enter**.

   The Technical Support Mode confirmation window opens.
5. Confirm that you want to operate the application in Technical Support Mode. To do so, select **Yes** and press **Enter**.
6. Run the following command:

   ```
   sudo -i
   ```
7. In the `/etc/sysconfig/apt-services` configuration file, in the `KAFKA_PORTS` field, delete the value `10000`.

If Secondary Central Node servers or the Sensor component installed on a separate server are connected to the Central Node server, you need to allow the connection with the server where you modified the configuration file via port 10000.

We do not recommend using this port for any external connections other than KUMA. To restrict connections over port 10000 only for KUMA, run the following command:

```
iptables -I INPUT -p tcp ! -s KUMA_IP_address --dport 10000 -j DROP
```

8. In the configuration file `/usr/bin/apt-start-sedr-iptables` add the value `10000` in the `WEB_PORTS` field, separated by a comma without a space.
9. Run the following command:

   ```
   sudo sh /usr/bin/apt-start-sedr-iptables
   ```

Preparations for exporting events on the Kaspersky Endpoint Detection and Response side are now complete.

In 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 file`—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 program component administrator menu is displayed.
3. In the program component administrator menu, select **Technical Support Mode**.
4. Press **Enter**.

   The Technical Support Mode confirmation window opens.
5. Confirm that you want to operate the application in Technical Support Mode. To do so, select **Yes** and press **Enter**.
6. In the `/usr/local/lib/python3.8/dist-packages/firewall/create_iptables_rules.py` configuration file, specify the additional port `10000` for the `WEB_PORTS` constant:

   ```
   WEB_PORTS = f'10000,80,{AppPort.APT_AGENT_PORT},{AppPort.APT_GUI_PORT}'
   ```

   `You do not need to perform this step for Kaspersky Endpoint Detection and Response 5.1 because the port is specified by default.`
7. Run the following commands:

   ```
   kata-firewall stop
   kata-firewall start --cluster-subnet <network mask for addressing cluster servers>
   ```

Preparations for exporting events on the Kaspersky Endpoint Detection and Response side are now complete.

In 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 file`—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 import Kaspersky Endpoint Detection and Response 5.1 events from hosts using the kata/edr connector:

1. Configure event receipt on the KUMA side. To do this, in KUMA, create and install a new 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 are start arriving in KUMA.

Creating 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", fill in the following fields:

   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. Select 'kata/edr' as the **Connector kind**. After you select the kata/edr connector type, more fields to fill in appear.

      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. Unpack the archive. Click **Upload certificate** and select the cert.pem file. Click **Upload private key** and select the key.pem file. Clicking **Create** adds the secret to the **Secret** drop-down list and automatically selects it.

      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. 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 KEDR Help. You can specify an arbitrary value that must not exceed the value on the KEDR side. If the value you specify exceeds the value of the **Maximum number of events** setting specified on the KEDR server, the KUMA collector log will display the error "Bad Request: max_events N is greater than the allowed value".

    d. The **Events fetch timeout** is set to 0 by default, which means that the default value configured on the KEDR server is applied. For details, refer to KEDR 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, refer to the KEDR Help.

4. At step 3 "Parsing", click **Add event parsing** and select "[OOTB] KEDR telemetry" in the **Basic event parsing** window.

5. To finish creating the collector in the web interface, click **Create and save service**. Then copy the collector installation command from the web interface and run this installation command on the command line on the server where you want to install the collector. If you were editing an existing collector, click **Save and restart services**.

As a result, the collector is created and is ready to send requests; the collector is displayed in the **Resources → 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.

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.

As a result, the integration is configured and you can view events arriving from KEDR in the KUMA → **Events** section.

Possible errors and solutions

If in the the collector log, you see the "Conflict: An external system with the following ip and certificate digest already exists. Either delete it or provide a new certificate" error, create a new secret with the a certificate in the connector of the collector.

If in the collector log, you see the "Continuation token not found" error in response to an event request, create a new connector, attach it to the collector and restart the collector; alternatively, create a new secret with a new certificate in the connector of the collector. If you do not want to receive events generated before the error occurred, configure a Timestamp filter in the collector.

Configuring the display of a link to a Kaspersky Endpoint Detection and Response detection in KUMA event details

When Kaspersky Endpoint Detection and Response detections are received, KUMA creates an alert for each detection. You can configure the display of a link to a Kaspersky Endpoint Detection and Response detection in KUMA alert information.

> You can configure the display of a detection link if you use only one Central Node server in Kaspersky Endpoint Detection and Response. If Kaspersky Endpoint Detection and Response is used in a distributed solution mode, it is impossible to configure the display of the links to Kaspersky Endpoint Detection and Response detections in KUMA.

To configure the display of a link to a detection in KUMA alert details, you need to complete steps in the Kaspersky Endpoint Detection and Response web interface and KUMA.

In the Kaspersky Endpoint Detection and Response web interface, you need to configure the integration of the application with KUMA as a SIEM system. For details on configuring integration, refer to the *Kaspersky Anti Targeted Attack Platform* documentation, *Configuring integration with a SIEM system* section.

Configuring the display of a link in the 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.

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 category list of the tenant by clicking + next to its name.

3. Select the required category or subcategory and click the **Add category** button.

   The **Add category** details area appears in the right part of the web interface window.

4. Define the category settings:

   a. In the **Name** field, enter the name of the category.

   b. In the **Parent** field, indicate the position of the category within the categories tree hierarchy. To do so, click the button ☰ and select a parent category for the category you are creating.

      Selected category appears in **Parent** fields.

   c. If required, define the values for the following settings:

      - Assign a severity to the category in the **Priority** drop-down list.

        The specified severity is assigned to correlation events and alerts associated with the asset.

      - If required, add a description for the category in the **Description** field.

      - 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. In the **Conditions** settings block, specify the filter for matching assets to attach to an asset category.

   You can add conditions by clicking the **Add condition** buttons. Groups of conditions can be added by using the **Add group** buttons. Group operators can be switched between **AND**, **OR**, and **NOT** values.

   Categorization filter operands and operators ?

Operand	Operators	Comment
Build number	>, >=, =, <=, <	
OS	=, like	The "like" operator ensures that the search is not case sensitive.
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	=, like	The "like" operator ensures that the search is not case sensitive.
CVE	=, in	The "in" operator lets you specify an array of values.
Software	=, like	
CII	in	More than one value can be selected.
Anti-virus databases last updated	>=,<=	
Last update of the information	>=,<=	
Protection last updated	>=,<=	
System last 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	>=,<=	

3. Use the **Test conditions** button to make sure that the specified filter is correct. When you click the button, you should see the **Assets for given conditions** window containing a list of assets that satisfy the search conditions.

- **Reactive**—the category will be filled with assets by using correlation rules.

5. Click 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 an asset name.

   b. In the **Tenant** drop-down list, select the tenant that will own the asset.

   c. In the **IP address** field, specify the IP address of the Kaspersky Endpoint Detection and Response Central Node server from which you want to receive detections.

   d. In the **Categories** field, select the category that you added in the previous step.

   > If you are using a predefined correlation rule, you need to select the `KATA standAlone` category.

   e. If required, define the values for the following fields:

   - In the **FQDN** field, specify the Fully Qualified Domain Name of the Kaspersky Endpoint Detection and Response server.

   - In the **MAC address** field, specify the MAC address of the Central Node Kaspersky Endpoint Detection and Response Central Node server.

   - In the **Owner** field, define the name of the asset owner.

4. Click the **Save** button.

Step 2. Adding a correlation rule

To add a correlation rule:

1. In the KUMA web interface, select the **Resources** section.
2. Select **Correlation rules** and click the **Create correlation rule** button.
3. On the **General** tab, specify the following settings:
   a. In the **Name** field, define the rule name.

   b. In the **Type** drop-down list, select **simple**.

   c. In the **Propagated fields** field, add the following fields: DeviceProduct, DeviceAddress, EventOutcome, SourceAssetID, DeviceAssetID.

   d. If required, define the values for the following fields:
   - In the **Rate limit** field, define the maximum number of times per second that the rule will be triggered.
   - In the **Severity** field, define the severity of alerts and correlation events that will be created as a result of the rule being triggered.
   - In the **Description** field, provide any additional information.
4. On the **Selectors** → **Settings** tab, specify the following settings:
   a. In the **Filter** drop-down list, select **Create new**.

   b. In the **Conditions** field, click the **Add group** button.

   c. In the operator field for the group you added, select **AND**.

   d. Add a condition for filtering by KATA value:
      1. In the **Conditions** field, click the **Add condition** button.
      2. In the condition field, select **If**.
      3. In the **Left operand** field, select **Event field**.
      4. In the **Event field** field, select **DeviceProduct**.
      5. In the **operator** field, select =.
      6. In the **Right operand** field, select **constant**.
      7. In the **value** field, enter KATA.

   e. Add a category filter condition:
      1. In the **Conditions** field, click the **Add condition** button.
      2. In the condition field, select **If**.
      3. In the **Left operand** field, select **Event field**.

        4. In the **Event field** field, select **DeviceAssetID**.

        5. In the **operator** field, select **inCategory**.

        6. In the **Right operand** field, select **constant**.

        7. Click the button.

        8. Select the category in which you placed the Kaspersky Endpoint Detection and Response Central Node server asset.

        9. Click the **Save** button.

    f. In the **Conditions** field, click the **Add group** button.

    g. In the operator field for the group you added, select **OR**.

    h. Add a condition for filtering by event class identifier:

        1. In the **Conditions** field, click the **Add condition** button.

        2. In the condition field, select **If**.

        3. In the **Left operand** field, select **Event field**.

        4. In the **Event field** field, select **DeviceEventClassID**.

        5. In the **operator** field, select =.

        6. In the **Right operand** field, select **constant**.

        7. In the **value** field, enter **taaScanning**.

    i. Repeat steps 1–7 in F for each of the following event class IDs:

        - file_web.

        - file_mail.

        - file_endpoint.

        - file_external.

        - ids.

        - url_web.

        - url_mail.

        - dns.

        - iocScanningEP.

        - yaraScanningEP.

5. On the **Actions** tab, specify the following settings:

a. In the **Actions** section, open the **On every event** drop-down list.

b. Select the **Output** check box.

c. In the **Enrichment** section, click the **Add enrichment** button.

d. In the **Source kind** drop-down list, select **template**.

e. In the **Template** field, enter *https://{{.DeviceAddress}}:8443/katap/#/alerts?id={{.EventOutcome}}*.

f. In the **Target field** drop-down list, select **DeviceExternalID**.

g. If necessary, turn on the **Debug** toggle switch to log information related to the operation of the resource.

6. Click the **Save** button.

Step 3. Creating a correlator

You need to launch the correlator installation wizard. At step 3 of the wizard, you are required to select the correlation rule that you added by following this guide.

After the correlator is created, a link to these detections will be displayed in the details of alerts created when receiving detections from Kaspersky Endpoint Detection and Response. The link is displayed in the correlation event details (**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.

  > CyberTrace web interface integration is available only if your CyberTrace license includes multi-user feature.

Integrating CyberTrace indicator search

To integrate CyberTrace indicator search:

1. Configure CyberTrace to receive and process KUMA requests.

You can configure the integration with KUMA immediately after installing CyberTrace in the Quick Start Wizard or later in the CyberTrace web interface.

2. Create an event enrichment rule in KUMA.

   In the enrichment rule, you can specify which data from CyberTrace you want to enrich the event with.

3. Create a collector to receive events that you want to enrich with CyberTrace data.

4. Link the enrichment rule to the collector.

5. Save and create the service:

   - If you linked the rule to a new collector, click **Save and create**, copy the collector ID in the opened window and use the copied ID to install the collector on the server using the command line interface.

   - If you linked the rule to an existing collector, click **Save and restart services** to apply the settings.

   The configuration of the integration of CyberTrace indicator search is complete and KUMA events will be enriched with CyberTrace data.

Example of testing CyberTrace data enrichment ?.

By default, KUMA does not test the connection with CyberTrace.

If you want to test the integration with CyberTrace and make sure that event enrichment is working, you can follow the steps of the following example or adapt the example to your situation. The example shows an integration test, which performs enrichment and shows that the event contains the specified test URL.

To run the test:

1. Create a test enrichment rule with parameters listed in the table below.

Setting	Value
Name	Test CT enrichment
Tenant	Shared
Source kind	CyberTrace
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 program web interface.

To configure CyberTrace to receive and process requests in the Quick Start Wizard:

1. Wait for the CyberTrace Quick Start Wizard to start after the program is installed.

   The **Welcome to Kaspersky CyberTrace** window opens.

2. In the **<select SIEM>** drop-down list, select the type of SIEM system from which you want to receive data and click the **Next** button.

   The **Connection Settings** window opens.

3. Do the following:

   a. In the **Service listens on** settings block, select the **IP and port** option.

   b. In the **IP address** field, enter `0.0.0.0`.

   c. In the **Port** field, enter the port for receiving events, the default port is `9999`.

   d. Under **Service sends events to**, specify `127.0.0.1` in the **IP address or hostname** field and in the **Port** field, specify `9998`.

      Leave the default values for everything else.

   e. Click **Next**.

   The **Proxy Settings** window opens.

4. If a proxy server is being used in your organization, define the settings for connecting to it. If not, leave all the fields blank and click **Next**.

   The **Licensing Settings** window opens.

5. In the **Kaspersky CyberTrace license key** field, add a license key for CyberTrace.

6. In the **Kaspersky Threat Data Feeds certificate** field, add a certificate that allows you to download updated data feeds from servers, and click **Next**.

CyberTrace will be configured.

To configure CyberTrace to receive and process requests in the program web interface:

1. In the CyberTrace web interface window, select **Settings** – **Service**.

2. In the **Connection Settings** block:

   a. Select the **IP and port** option.

   b. In the **IP address** field, enter `0.0.0.0`.

   c. In the **Port** field, specify the port for receiving events, the default port is `9999`.

3. In the **Web interface** settings block, in the **IP address or hostname** field, enter `127.0.0.1`.

4. In the upper toolbar, click **Restart the CyberTrace Service**.

5. Select **Settings** – **Events format**.

6. In the **Alert events format** field, enter `%Date% alert=%Alert%%RecordContext%`.

7. In the **Detection events format** field, enter `Category=%Category%|MatchedIndicator=%MatchedIndicator%%RecordContext%`.

8. In the **Records context format** field, enter `|%ParamName%=%ParamValue%`.

9. In the **Actionable fields context format** field, enter `%ParamName%:%ParamValue%`.

CyberTrace will be configured.

> After updating CyberTrace configuration you have to restart the CyberTrace server.

Creating event Enrichment rules

To create event enrichment rules:

1. In the KUMA web interface, open the **Resources** → **Enrichment rules** section and in the left part of the window, select or create a folder for the new rule.

   The list of available enrichment rules will be displayed.

2. Click **Add enrichment rule** to create a new rule.

   The enrichment rule window will be displayed.

3. Enter the rule configuration parameters:

   a. In the **Name** field, enter a unique name for the rule. The name must contain 1 to 128 Unicode characters.

   b. In the **Tenant** drop-down list, select the tenant that will own this resource.

   c. In the **Source kind** drop-down list, select **cybertrace**.

   d. Specify the **URL** of the CyberTrace server to which you want to connect. For example, *example.domain.com:9999*.

e. If necessary, use the **Number of connections** field to specify the maximum number of connections to the CyberTrace server that can be simultaneously established by KUMA. By default, this value is equal to the number of vCPUs of the KUMA Core server.

f. In the **RPS** field, enter the number of requests to the CyberTrace server per second that KUMA can make. The default value is `1000`.

g. 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. The default value is `30`.

h. In the **Mapping** settings block, you must specify the fields of events to be checked via CyberTrace, and define the rules for mapping fields of KUMA events to CyberTrace indicator types:

- In the **KUMA field** column, select the field whose value must be sent to CyberTrace.
- In the **CyberTrace indicator** column, select the CyberTrace indicator type for every field you selected:
  - **ip**
  - **url**
  - **hash**

You must provide at least one string to the table. You can use the **Add row** button to add a string, and can use the × button to remove a string.

i. Use the **Debug** toggle switch to indicate whether or not to enable logging of service operations. Logging is disabled by default.

j. If necessary, in the **Description** field, add up to 4,000 Unicode characters describing the resource.

k. 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

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. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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**.

   A new enrichment rule will be created.

CyberTrace indicator search integration is now configured. You can now add the created enrichment rule to a collector. You must restart KUMA collectors to apply the new settings.

If any of the CyberTrace fields in the events details area contains "[{" or "}]" values, it means that information from CyberTrace data feed was processed incorrectly and it's possible that some of the data is not displayed. You can get all data feed information by copying the events **TI indicator** field value from KUMA and searching for it in the CyberTrace in the indicators section. All relevant information will be displayed in the **Indicator context** section of CyberTrace.

Integrating CyberTrace interface

You can integrate the CyberTrace web interface 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 **Settings** → **Kaspersky CyberTrace** section of the KUMA web interface.

To integrate the CyberTrace web interface in KUMA:

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. This resource is used to store credentials of the CyberTrace server.

   The secret window is displayed.

3. Enter information about the secret:

a. In the **Name** field, choose a name for the added secret. The name must contain 1 to 128 Unicode characters.

b. In the **Tenant** drop-down list, select the tenant that will own this resource.

c. In the **Type** drop-down list, select **credentials**.

d. In the **User** and **Password** fields, enter credentials for your CyberTrace server.

e. If necessary, in the **Description** field, add up to 4,000 Unicode characters describing the resource.

4. Click **Save**.

   The CyberTrace server credentials are now saved and can be used in other KUMA resources.

5. In the KUMA web interface, open **Settings** → **Kaspersky CyberTrace**.

   The window with CyberTrace integration parameters opens.

6. Make the necessary changes to the following parameters:

   - **Disabled**—clear this check box if you want to integrate 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.

7. In the **Secret** drop-down list, select the secret you created before.

8. 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 work with the program web interface, the CyberTrace section may fail to display data. In this case, configure the data display (see below).

9. Click **Save**.

CyberTrace is now integrated with 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**.

The selected object is now added to the CyberTrace denylist.

Integration with Kaspersky Threat Intelligence Portal

The Kaspersky Threat Intelligence Portal combines all of Kaspersky's knowledge about cyberthreats and how they're related into a single web service. When integrated with KUMA, it helps KUMA users to make faster and better-informed decisions, providing them with data about URLs, domains, IP addresses, WHOIS / DNS data.

> Access to the Kaspersky Threat Intelligence Portal is provided based on a fee. License certificates are created by Kaspersky experts. To obtain a certificate for Kaspersky Threat Intelligence Portal, contact your Technical Account Manager.

Initializing integration

To integrate Kaspersky Threat Intelligence Portal into KUMA:

1. In the 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. This resource is used to store credentials of your Kaspersky Threat Intelligence Portal account.

   The secret window is displayed.

3. Enter information about the secret:

   a. In the **Name** field, choose a name for the added secret.

   b. In the **Tenant** drop-down list, select the tenant that will own the created resource.

   c. In the **Type** drop-down list, select **ktl**.

   d. In the **User** and **Password** fields, enter credentials for your Kaspersky Threat Intelligence Portal account.

   e. If you want, enter a **Description** of the secret.

4. Upload your Kaspersky Threat Intelligence Portal certificate key:

   a. Click the **Upload PFX** button and select the PFX file with your certificate.

      The name of the selected file appears to the right of the **Upload PFX** button.

   b. Enter the password to the PFX file in the **PFX password** field.

5. Click **Save**.

   The Kaspersky Threat Intelligence Portal account credentials are now saved and can be used in other KUMA resources.

6. In the **Settings** section of the KUMA web interface, open the **Kaspersky Threat Lookup** tab.

   The list of available connections will be displayed.

7. Make sure the **Disabled** check box is cleared.

8. 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.

9. If necessary, select a proxy server in the **Proxy** drop-down list.

10. Click **Save**.

11. 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 which KUMA has information available, the data from Kaspersky Threat Intelligence Portal opens, with the time it was received indicated at the bottom, instead of the **Threat Lookup enrichment** window. You can update the data.

Updating information from Kaspersky Threat Intelligence Portal

To update information, received from Kaspersky Threat Intelligence Portal:

1. Open the event details area from the events table, alert window, or correlation event window and click the link on a domain, web address, IP address, or file hash for which you previously requested information from Kaspersky Threat Intelligence Portal.

2. Click **Update** in the event details area containing the data received from the Kaspersky Threat Intelligence Portal.

   The **Threat Lookup enrichment** area opens in the right part of the screen.

3. Select the check boxes next to the types of information you want to request.

   If neither check box is selected, all information types are requested.

4. In the **Maximum number of records in each data group** field enter the number of entries per selected information type you want to receive. The default value is `10`.

5. Click **Update**.

   The *KTL* task is created and the new data received from Kaspersky Threat Intelligence Portal is requested.

6. Close the **Threat Lookup enrichment** window and the details area with KTL information.

7. Open the event details area from the events table, Alert window or correlation event window and click the link on a domain, URL, IP address, or file hash for which you updated Kaspersky Threat Intelligence Portal information and select **Show info from Threat Lookup**.

The event details area opens on the right with data from Kaspersky Threat Intelligence Portal, indicating the time when it was received on the bottom of the screen.

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 applications. 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** → **IRP / SOAR**.

To configure integration with R-Vision SOAR:

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. 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** → **General** → **API**.

e. If necessary, in the **Description** field, add up to 4,000 Unicode characters describing the secret.

4. Click **Save**.

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 **Settings** → **IRP / SOAR**.

The window containing R-Vision SOAR integration settings opens.

6. Make the necessary changes to the following parameters:

- **Disabled**—select this check box if you want to disable R-Vision SOAR integration with KUMA.

- In the **Secret** drop-down list, select the previously created secret.

  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.

- **URL** (required)—URL of the R-Vision SOAR server host.

- **Field name where KUMA alert IDs must be placed** (required)—name of the R-Vision SOAR field where the ID of the KUMA alert must be written.

- **Field name where KUMA alert URLs must be placed** (required)—name of the R-Vision SOAR field where the link for accessing the KUMA alert should be written.

- **Category** (required)—category of R-Vision SOAR incident that is created after KUMA alert is received.

- **KUMA event fields that must be sent to IRP / SOAR** (required)—drop-down list for selecting the KUMA event fields that should be sent to R-Vision SOAR.

- **Severity** group of settings (required)—used to map KUMA severity values to R-Vision SOAR severity values.

7. 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.

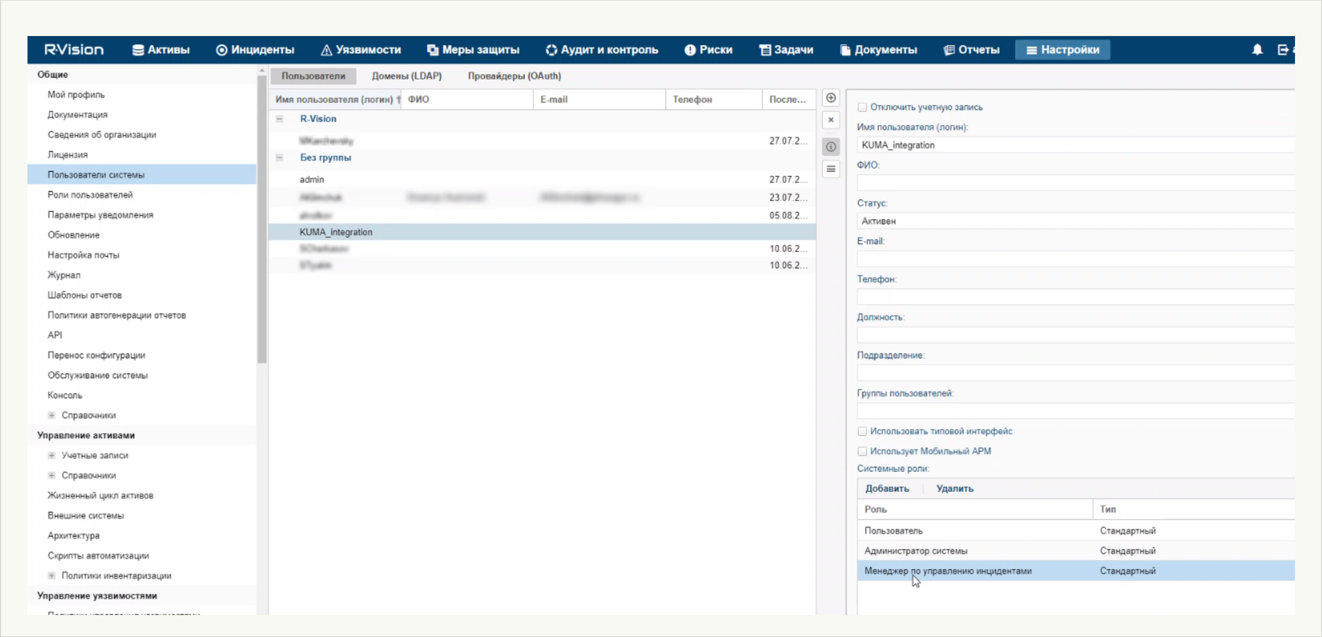
Integration in R-Vision SOAR is configured in the **Settings** section of the R-Vision SOAR web interface. For details on configuring R-Vision SOAR, please refer to the documentation on this application.

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** → **General** → **System users** section. The role is added in the **System Roles** block of settings.

     R-Vision SOAR version 4.0 user with the Incident Manager role

Имя пользователя (логин)	ФИО	E-mail	Телефон	После...
R-Vision				
				27.07.2...
Без группы				
admin				27.07.2...
				23.07.2...
				05.08.2...
KUMA_integration				
				10.06.2...
				10.06.2...

     ☐ Отключить учетную запись

     Имя пользователя (логин): KUMA_integration

     ФИО:

     Статус: Активен

     E-mail:

     Телефон:

     Должность:

     Подразделение:

     Группы пользователей:

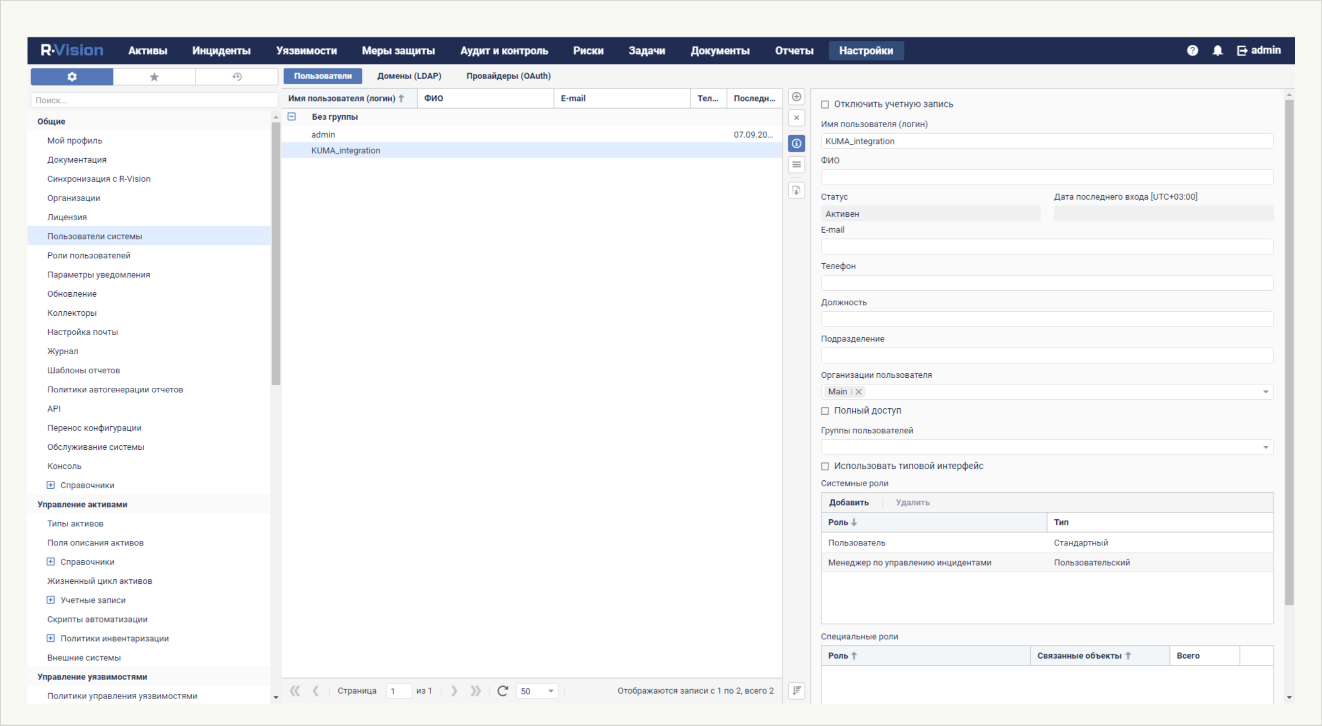
     ☐ Использовать типовой интерфейс

     ☐ Использует Мобильный АРМ

     Системные роли:

Роль	Тип
Пользователь	Стандартный
Администратор системы	Стандартный
Менеджер по управлению инцидентами	Стандартный

     R-Vision SOAR version 5.0 user with the Incident Manager role

Имя пользователя (логин)	ФИО	E-mail	Тел...	Последн...
Без группы				
admin				07.09.20...
KUMA_integration				

     ☐ Отключить учетную запись

     Имя пользователя (логин): KUMA_integration

     ФИО

     Статус: Активен

     Дата последнего входа [UTC+03:00]

     E-mail

     Телефон

     Должность

     Подразделение

     Организации пользователя: Main

     ☐ Полный доступ

     Группы пользователей

     ☐ Использовать типовой интерфейс

     Системные роли

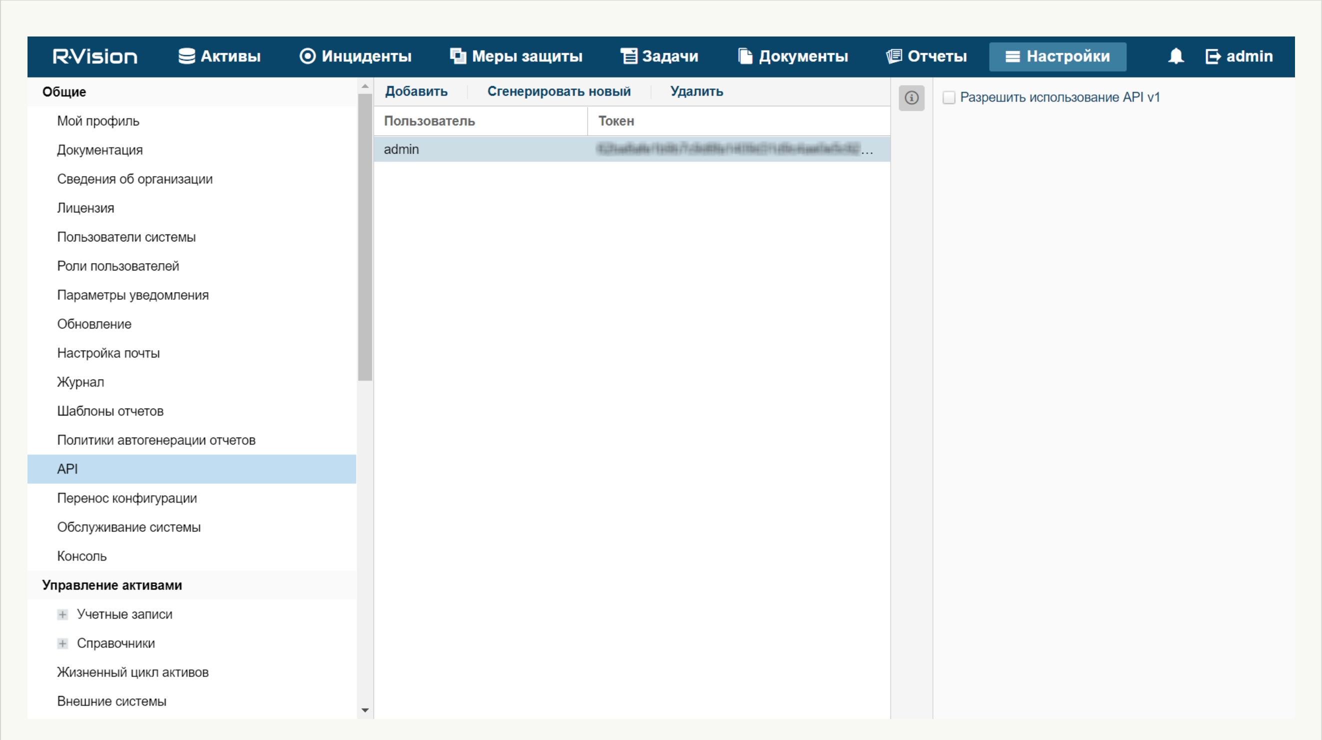
Роль	Тип
Пользователь	Стандартный
Менеджер по управлению инцидентами	Пользовательский

     Специальные роли

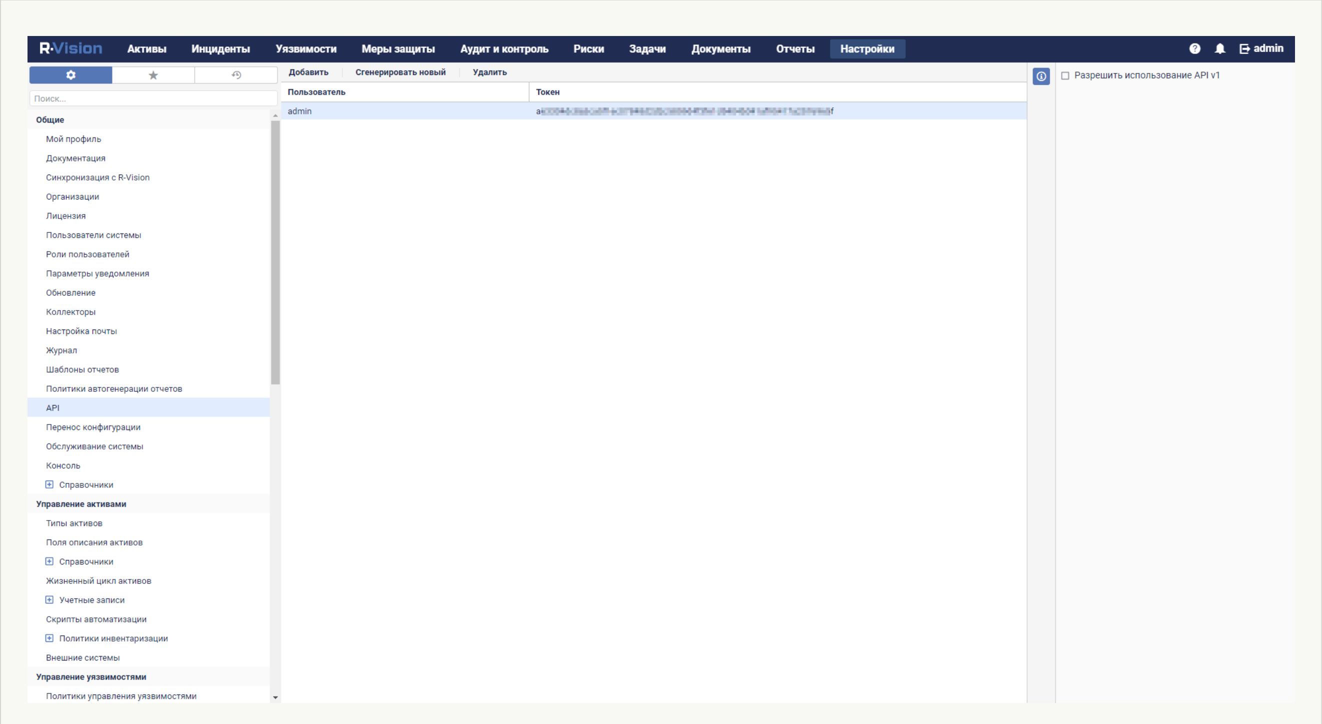
Роль	Связанные объекты	Всего

  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** → **General** → **API**.

     API token in R-Vision SOAR version 4.0



[API token in R-Vision SOAR version 5.0](#)

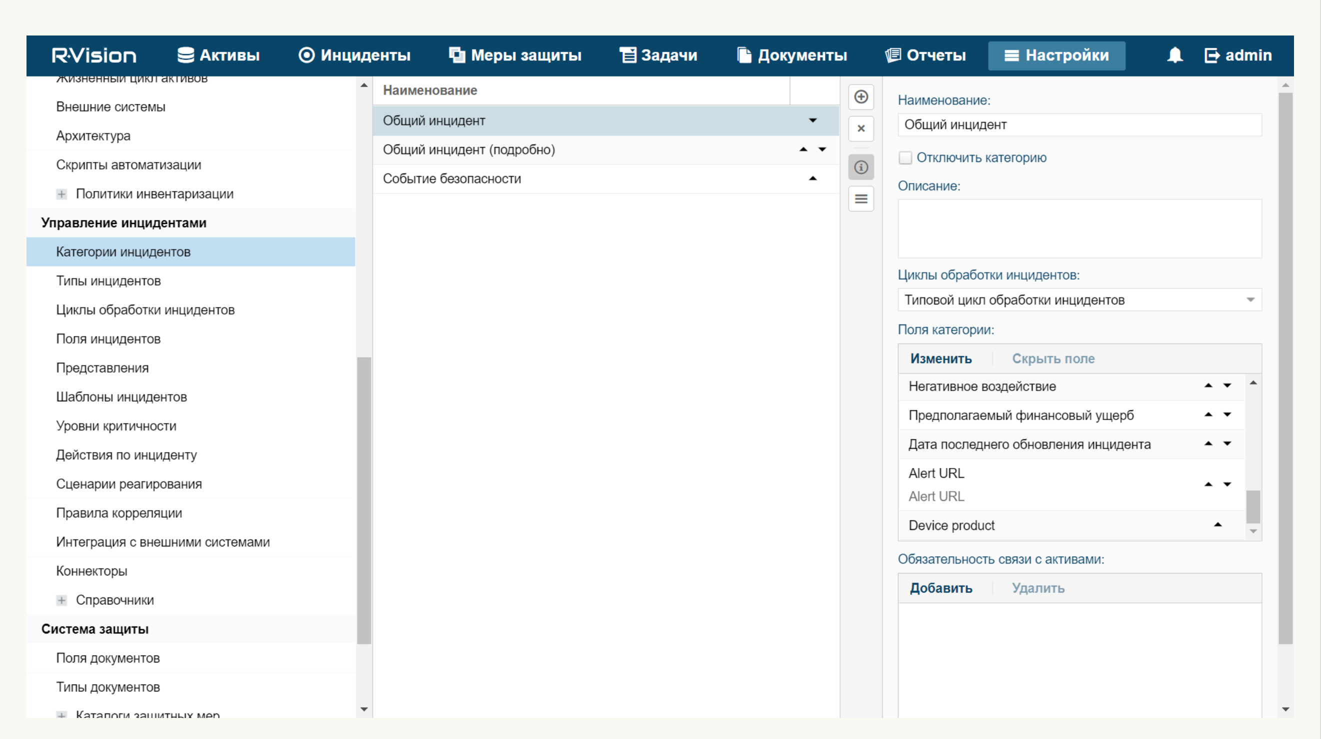


- **Configuring R-Vision SOAR incident fields and KUMA alert fields**

  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 in the **Category fields** settings block. The `Alert ID` field can be hidden.

     [Incident categories with data from KUMA alerts in R-Vision SOAR version 4.0](#)



Жизненный цикл активов
Внешние системы
Архитектура
Скрипты автоматизации
Политики инвентаризации
Управление инцидентами
Категории инцидентов
Типы инцидентов
Циклы обработки инцидентов
Поля инцидентов
Представления
Шаблоны инцидентов
Уровни критичности
Действия по инциденту
Сценарии реагирования
Правила корреляции
Интеграция с внешними системами
Коннекторы
Справочники
Система защиты
Поля документов
Типы документов
Каталоги защитных мер

Наименование
Общий инцидент
Общий инцидент (подробно)
Событие безопасности

Наименование: Общий инцидент

□ Отключить категорию

Описание:

Циклы обработки инцидентов: Типовой цикл обработки инцидентов

Поля категории:

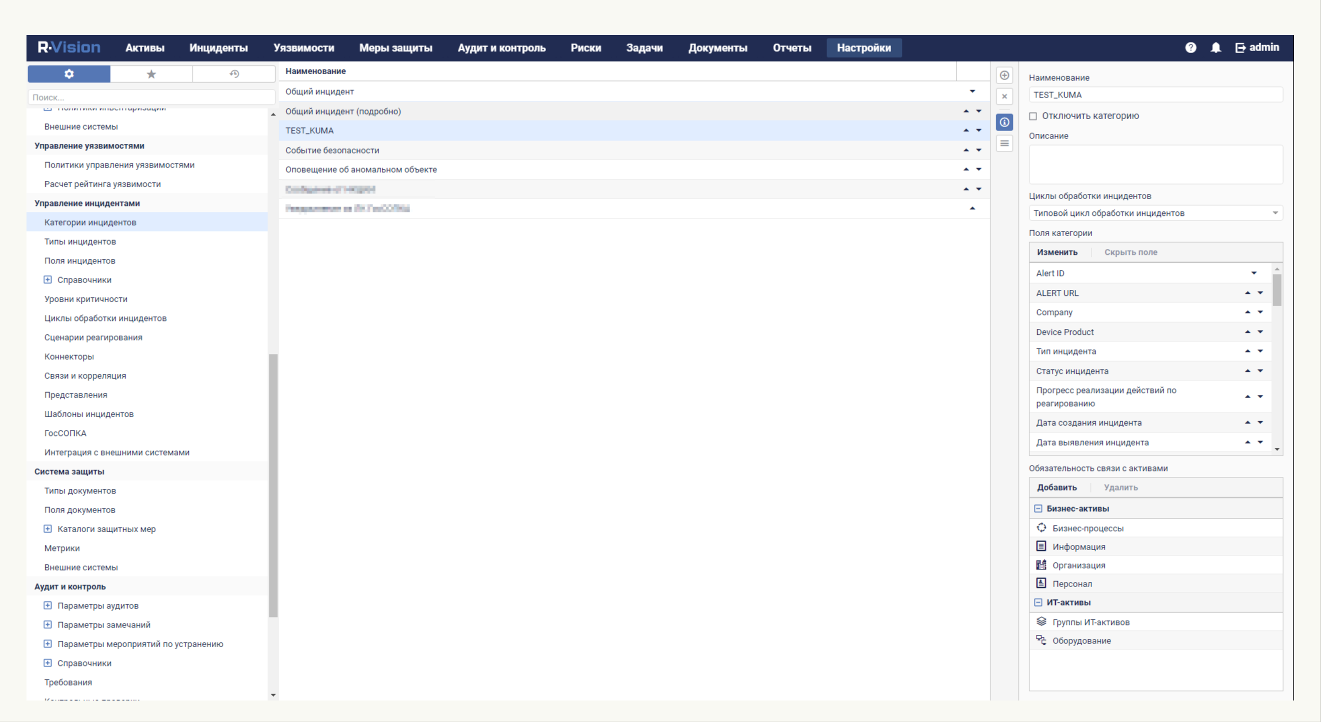
Изменить | Скрыть поле

- Негативное воздействие
- Предполагаемый финансовый ущерб
- Дата последнего обновления инцидента
- Alert URL
  Alert URL
- Device product

Обязательность связи с активами:

Добавить | Удалить

[Incident categories with data from KUMA alerts in R-Vision SOAR version 5.0](#) ?



Политики инвентаризации
Внешние системы
Управление уязвимостями
Политики управления уязвимостями
Расчет рейтинга уязвимости
Управление инцидентами
Категории инцидентов
Типы инцидентов
Поля инцидентов
Справочники
Уровни критичности
Циклы обработки инцидентов
Сценарии реагирования
Коннекторы
Связи и корреляция
Представления
Шаблоны инцидентов
ГосСОПКА
Интеграция с внешними системами
Система защиты
Типы документов
Поля документов
Каталоги защитных мер
Метрики
Внешние системы
Аудит и контроль
Параметры аудитов
Параметры замечаний
Параметры мероприятий по устранению
Справочники
Требования
Контрольные проверки

Наименование
Общий инцидент
Общий инцидент (подробно)
TEST_KUMA
Событие безопасности
Оповещение об аномальном объекте
[redacted]
[redacted]

Наименование: TEST_KUMA

□ Отключить категорию

Описание

Циклы обработки инцидентов: Типовой цикл обработки инцидентов

Поля категории

Изменить | Скрыть поле

- Alert ID
- ALERT URL
- Company
- Device Product
- Тип инцидента
- Статус инцидента
- Прогресс реализации действий по реагированию
- Дата создания инцидента
- Дата выявления инцидента

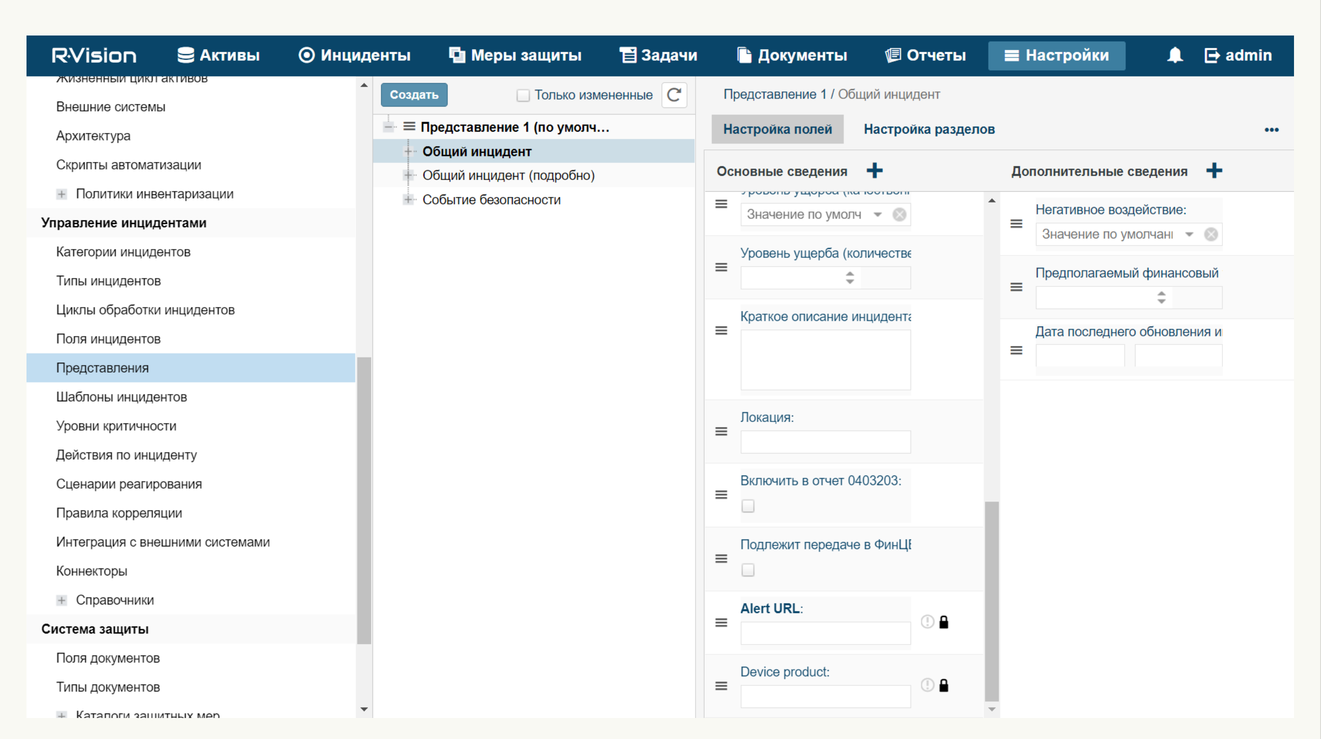
Обязательность связи с активами

Добавить | Удалить

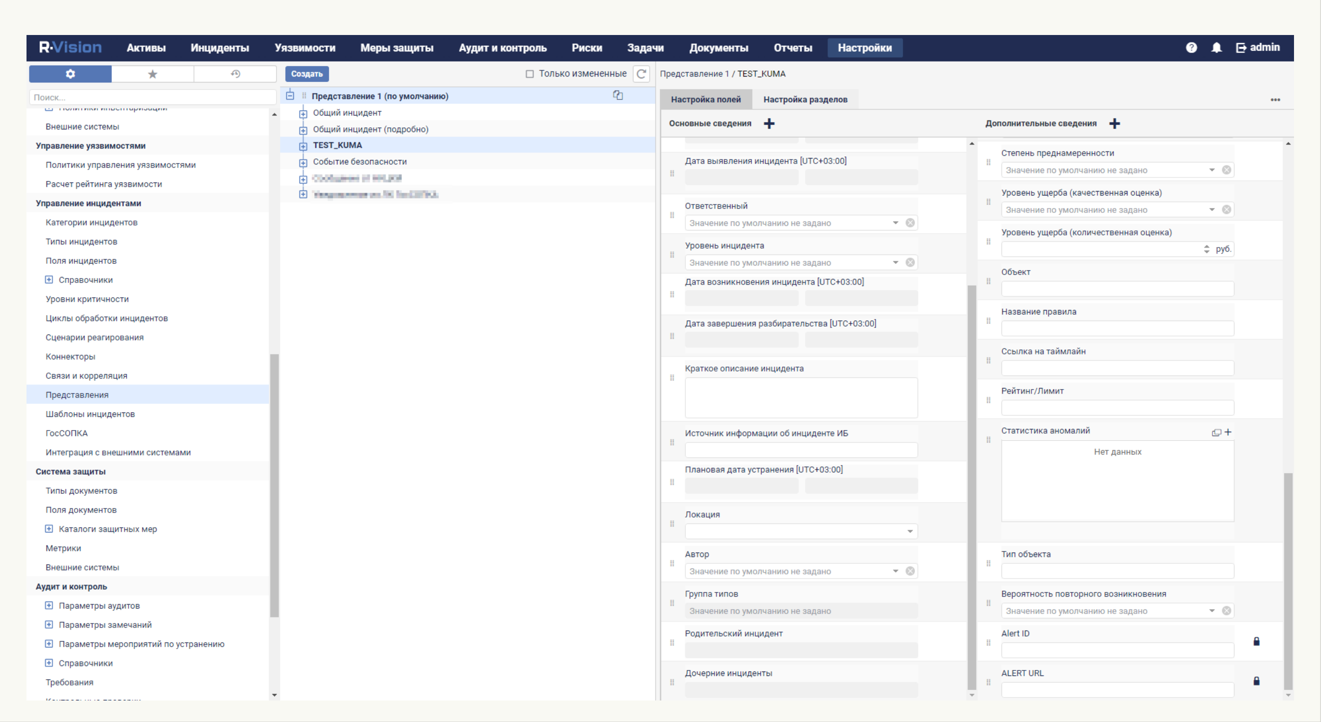
- **Бизнес-активы**
  - Бизнес-процессы
  - Информация
  - Организация
  - Персонал
- **ИТ-активы**
  - Группы ИТ-активов
  - Оборудование

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.

   [The Alert URL field is not editable in R-Vision SOAR version 4.0](#) ?



[The Alert URL field is not editable in R-Vision SOAR version 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

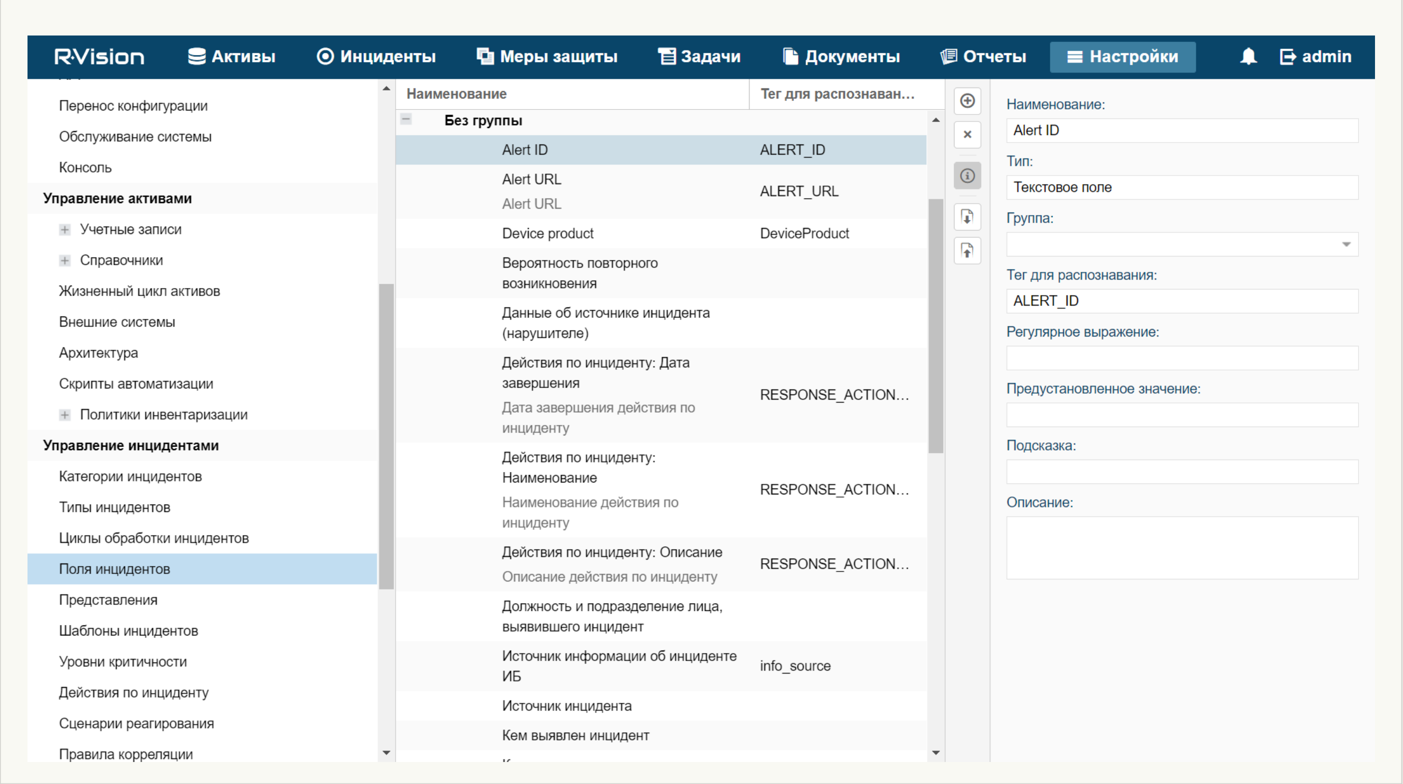
To add the ALERT_ID incident field in the R-Vision SOAR:

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`.

ALERT_ID field added to R-Vision SOAR incident.

ALERT_ID field in R-Vision SOAR version 4.0



Наименование	Тег для распознаван...
Без группы	
Alert ID	ALERT_ID
Alert URL Alert URL	ALERT_URL
Device product	DeviceProduct
Вероятность повторного возникновения	
Данные об источнике инцидента (нарушителе)	
Действия по инциденту: Дата завершения Дата завершения действия по инциденту	RESPONSE_ACTION...
Действия по инциденту: Наименование Наименование действия по инциденту	RESPONSE_ACTION...
Действия по инциденту: Описание Описание действия по инциденту	RESPONSE_ACTION...
Должность и подразделение лица, выявившего инцидент	
Источник информации об инциденте ИБ	info_source
Источник инцидента	
Кем выявлен инцидент	

Наименование: Alert ID

Тип: Текстовое поле

Группа:

Тег для распознавания: ALERT_ID

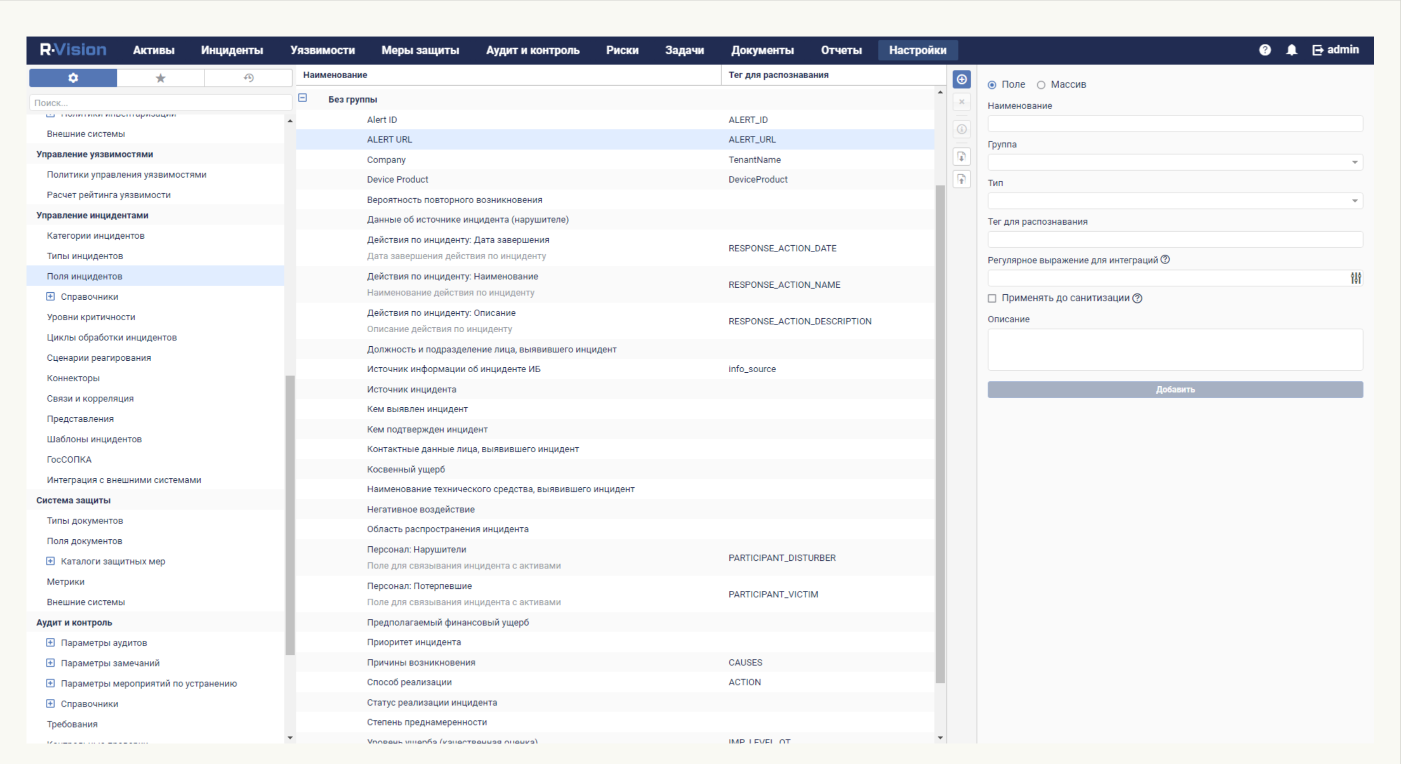
Регулярное выражение:

Предустановленное значение:

Подсказка:

Описание:

ALERT_ID field in R-Vision SOAR version 5.0



Наименование	Тег для распознавания
Без группы	
Alert ID	ALERT_ID
ALERT URL	ALERT_URL
Company	TenantName
Device Product	DeviceProduct
Вероятность повторного возникновения	
Данные об источнике инцидента (нарушителе)	
Действия по инциденту: Дата завершения Дата завершения действия по инциденту	RESPONSE_ACTION_DATE
Действия по инциденту: Наименование Наименование действия по инциденту	RESPONSE_ACTION_NAME
Действия по инциденту: Описание Описание действия по инциденту	RESPONSE_ACTION_DESCRIPTION
Должность и подразделение лица, выявившего инцидент	
Источник информации об инциденте ИБ	info_source
Источник инцидента	
Кем выявлен инцидент	
Кем подтвержден инцидент	
Контактные данные лица, выявившего инцидент	
Косвенный ущерб	
Наименование технического средства, выявившего инцидент	
Негативное воздействие	
Область распространения инцидента	
Персонал: Нарушители Поле для связывания инцидента с активами	PARTICIPANT_DISTURBER
Персонал: Потерпевшие Поле для связывания инцидента с активами	PARTICIPANT_VICTIM
Предполагаемый финансовый ущерб	
Приоритет инцидента	
Причины возникновения	CAUSES
Способ реализации	ACTION
Статус реализации инцидента	
Степень преднамеренности	
Уровень ущерба (качественная оценка)	IMP_LEVEL_QT

● Поле ○ Массив

Наименование

Группа

Тип

Тег для распознавания

Регулярное выражение для интеграций

□ Применять до санитизации

Описание

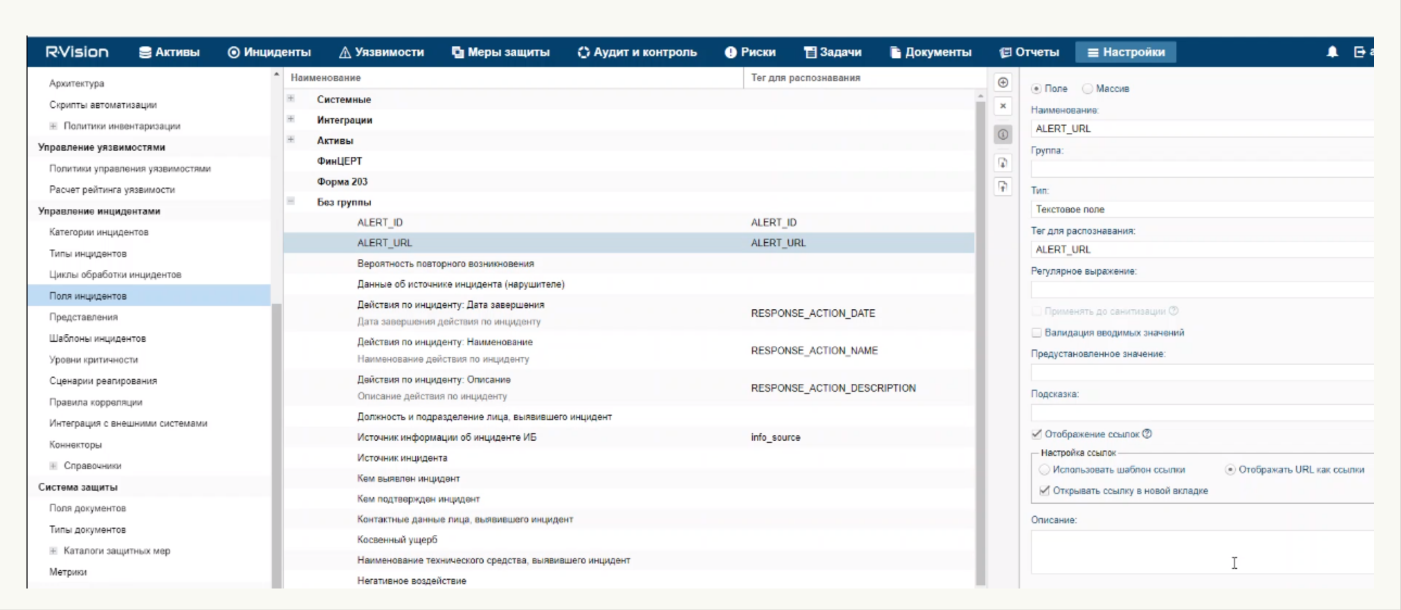
To add the ALERT_URL incident field in R-Vision SOAR:

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.

ALERT_URL field added to R-Vision SOAR incident.

[ALERT_URL field in R-Vision SOAR version 4.0](#)



Архитектура
Скрипты автоматизации
⊞ Политики инвентаризации
Управление уязвимостями
Политики управления уязвимостями
Расчет рейтинга уязвимости
Управление инцидентами
Категории инцидентов
Типы инцидентов
Циклы обработки инцидентов
Поля инцидентов
Представления
Шаблоны инцидентов
Уровни критичности
Сценарии реагирования
Правила корреляции
Интеграция с внешними системами
Коннекторы
⊞ Справочники
Система защиты
Поля документов
Типы документов
⊞ Каталоги защитных мер
Метрики

Наименование	Тег для распознавания
⊞ **Системные**	
⊞ **Интеграции**	
⊞ **Активы**	
ФинЦЕРТ	
Форма 203	
⊟ **Без группы**	
ALERT_ID	ALERT_ID
ALERT_URL	ALERT_URL
Вероятность повторного возникновения	
Данные об источнике инцидента (нарушителе)	
Действия по инциденту: Дата завершения Дата завершения действия по инциденту	RESPONSE_ACTION_DATE
Действия по инциденту: Наименование Наименование действия по инциденту	RESPONSE_ACTION_NAME
Действия по инциденту: Описание Описание действия по инциденту	RESPONSE_ACTION_DESCRIPTION
Должность и подразделение лица, выявившего инцидент	
Источник информации об инциденте ИБ	info_source
Источник инцидента	
Кем выявлен инцидент	
Кем подтвержден инцидент	
Контактные данные лица, выявившего инцидент	
Косвенный ущерб	
Наименование технического средства, выявившего инцидент	
Негативное воздействие	

● Поле ○ Массив

Наименование: ALERT_URL

Группа:

Тип: Текстовое поле

Тег для распознавания: ALERT_URL

Регулярное выражение:

□ Применять до санитизации

□ Валидация вводимых значений

Предустановленное значение:

Подсказка:

☑ Отображение ссылок

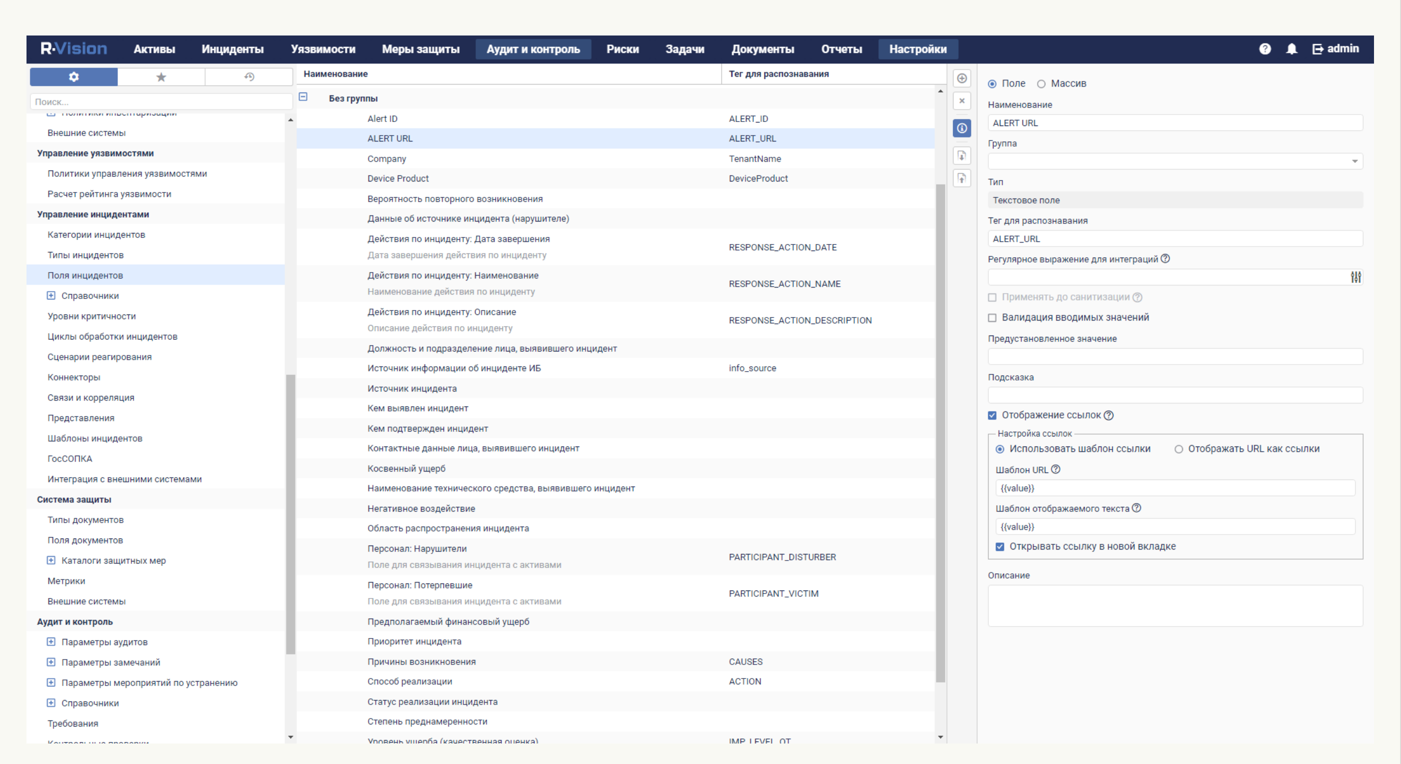
Настройка ссылок

○ Использовать шаблон ссылки ● Отображать URL как ссылки

☑ Открывать ссылку в новой вкладке

Описание:

ALERT_URL field in R-Vision SOAR version 5.0



⊞ Политики инвентаризации
Внешние системы
Управление уязвимостями
Политики управления уязвимостями
Расчет рейтинга уязвимости
Управление инцидентами
Категории инцидентов
Типы инцидентов
Поля инцидентов
⊞ Справочники
Уровни критичности
Циклы обработки инцидентов
Сценарии реагирования
Коннекторы
Связи и корреляция
Представления
Шаблоны инцидентов
ГосСОПКА
Интеграция с внешними системами
Система защиты
Типы документов
Поля документов
⊞ Каталоги защитных мер
Метрики
Внешние системы
Аудит и контроль
⊞ Параметры аудитов
⊞ Параметры замечаний
⊞ Параметры мероприятий по устранению
⊞ Справочники
Требования
Контрольные проверки

Наименование	Тег для распознавания
⊟ **Без группы**	
Alert ID	ALERT_ID
ALERT URL	ALERT_URL
Company	TenantName
Device Product	DeviceProduct
Вероятность повторного возникновения	
Данные об источнике инцидента (нарушителе)	
Действия по инциденту: Дата завершения Дата завершения действия по инциденту	RESPONSE_ACTION_DATE
Действия по инциденту: Наименование Наименование действия по инциденту	RESPONSE_ACTION_NAME
Действия по инциденту: Описание Описание действия по инциденту	RESPONSE_ACTION_DESCRIPTION
Должность и подразделение лица, выявившего инцидент	
Источник информации об инциденте ИБ	info_source
Источник инцидента	
Кем выявлен инцидент	
Кем подтвержден инцидент	
Контактные данные лица, выявившего инцидент	
Косвенный ущерб	
Наименование технического средства, выявившего инцидент	
Негативное воздействие	
Область распространения инцидента	
Персонал: Нарушители Поле для связывания инцидента с активами	PARTICIPANT_DISTURBER
Персонал: Потерпевшие Поле для связывания инцидента с активами	PARTICIPANT_VICTIM
Предполагаемый финансовый ущерб	
Приоритет инцидента	
Причины возникновения	CAUSES
Способ реализации	ACTION
Статус реализации инцидента	
Степень преднамеренности	
Уровень ущерба (качественная оценка)	IMP_LEVEL_OT

● Поле ○ Массив

Наименование

ALERT URL

Группа

Тип

Текстовое поле

Тег для распознавания

ALERT_URL

Регулярное выражение для интеграций

□ Применять до санитизации

□ Валидация вводимых значений

Предустановленное значение

Подсказка

☑ Отображение ссылок

Настройка ссылок

● Использовать шаблон ссылки ○ Отображать URL как ссылки

Шаблон URL

{{value}}

Шаблон отображаемого текста

{{value}}

☑ Открывать ссылку в новой вкладке

Описание

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 type API request to close an alert in the format `<KUMA Core server FQDN>:<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:

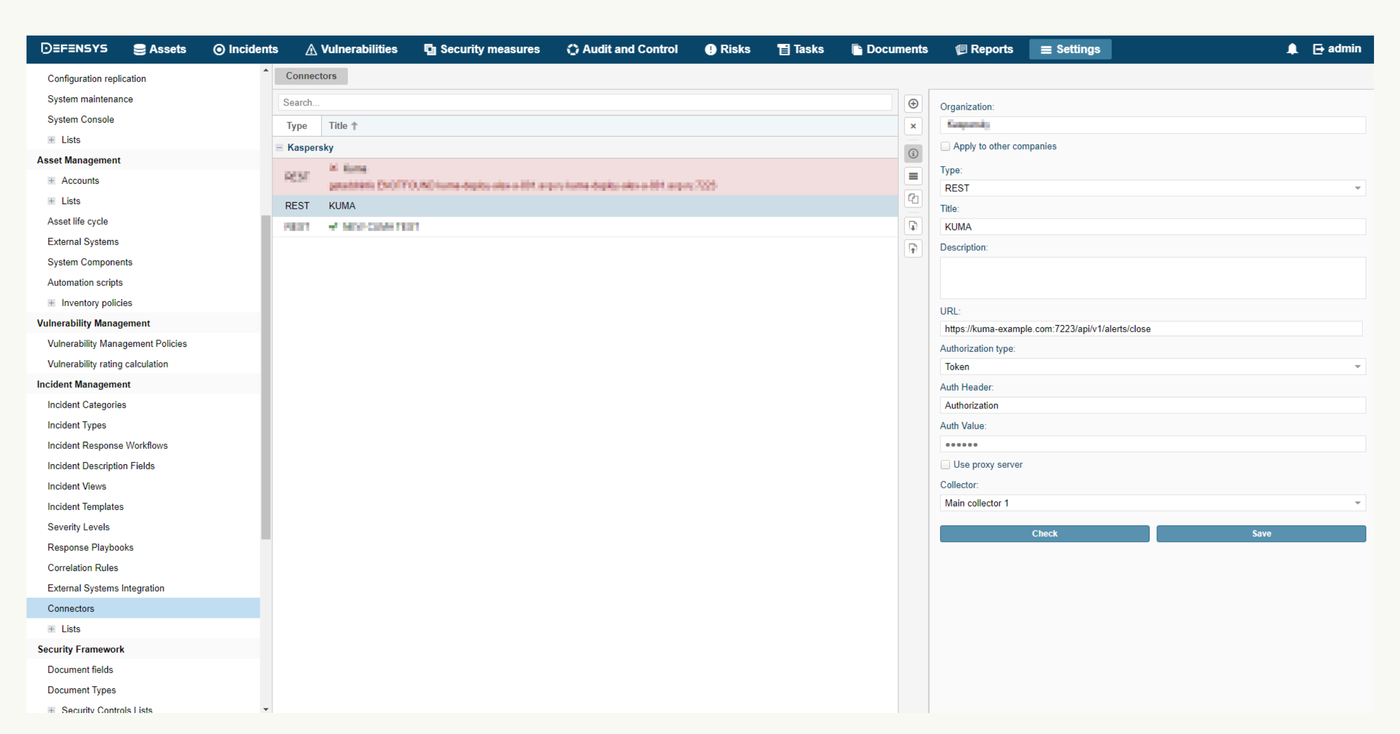
   ```
   Bearer <KUMA General administrator token>
   ```

8. In the **Collector** drop-down list select previously created collector.

9. Click **Save**.

The connector has been created.

Connector in R-Vision SOAR version 4.0



Type	Title
Kaspersky	
[redacted]	[redacted]
REST	KUMA
[redacted]	[redacted]

Organization: [redacted]

□ Apply to other companies

Type: REST

Title: KUMA

Description:

URL: https://kuma-example.com:7223/api/v1/alerts/close

Authorization type: Token

Auth Header: Authorization

Auth Value: ••••••

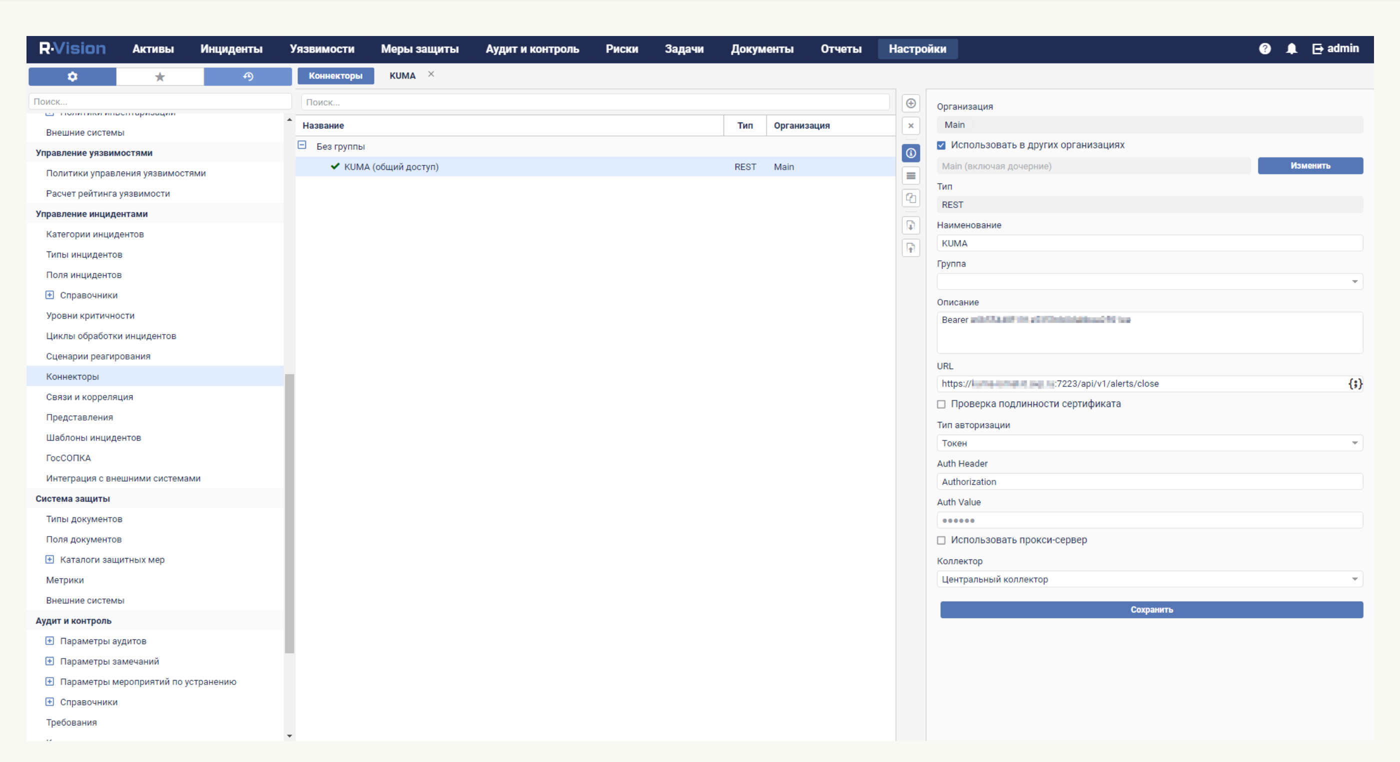
□ Use proxy server

Collector: Main collector 1

Check

Save

Connector in R-Vision SOAR version 5.0



Название	Тип	Организация
Без группы		
✓ KUMA (общий доступ)	REST	Main

Организация: Main

☑ Использовать в других организациях

Main (включая дочерние) Изменить

Тип: REST

Наименование: KUMA

Группа:

Описание: Bearer [redacted]

URL: https://[redacted]:7223/api/v1/alerts/close

□ Проверка подлинности сертификата

Тип авторизации: Токен

Auth Header: Authorization

Auth Value: ••••••

□ Использовать прокси-сервер

Коллектор: Центральный коллектор

Сохранить

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 type API request to close an alert in the format `<KUMA Core server FQDN>:<Port used for API requests (7223 by default)>/api/v1/alerts/close`.

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 contents of the API request body:

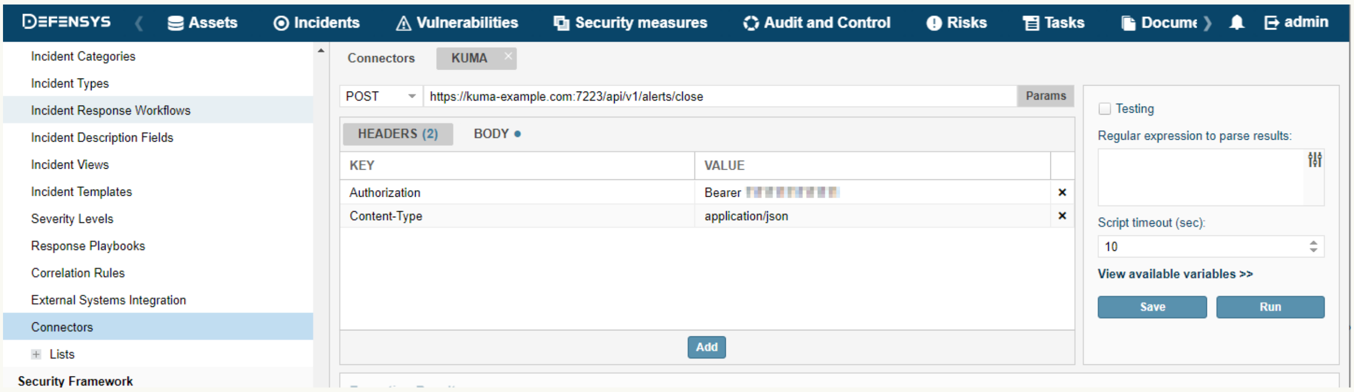
```
{
    "id":"{{tag.ALERT_ID}}",
    "reason":"<Reason for closing the alert. Available values: "Incorrect Correlation Rule", "Incorrect Data", "Responded".> "
}
```

6. Click **Save**.

The connector is configured.

Connector in R-Vision SOAR version 4.0

API request header

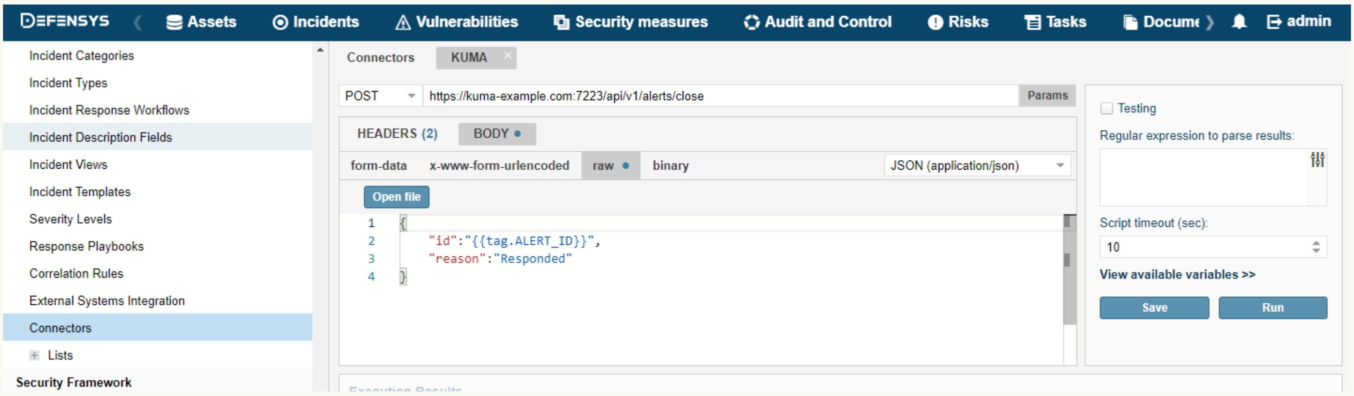


POST https://kuma-example.com:7223/api/v1/alerts/close

KEY	VALUE
Authorization	Bearer [redacted]
Content-Type	application/json

Script timeout (sec): 10

API request body



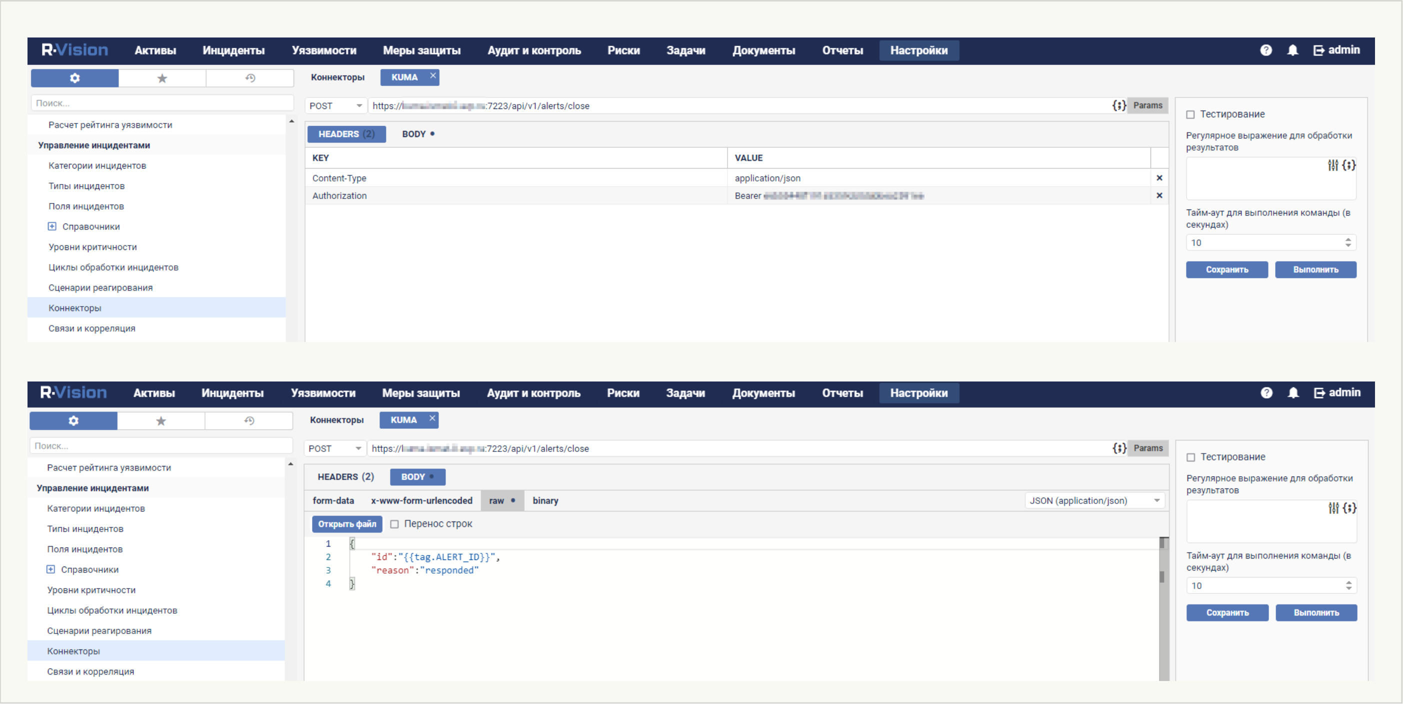
POST https://kuma-example.com:7223/api/v1/alerts/close

raw: JSON (application/json)

```
{
    "id":"{{tag.ALERT_ID}}",
    "reason":"Responded"
}
```

Script timeout (sec): 10

Connector in R-Vision SOAR version 5.0



POST https://[redacted]:7223/api/v1/alerts/close

HEADERS (2) | BODY •

KEY	VALUE
Content-Type	application/json
Authorization	Bearer [redacted]

□ Тестирование

Регулярное выражение для обработки результатов

Тайм-аут для выполнения команды (в секундах): 10

POST https://[redacted]:7223/api/v1/alerts/close

HEADERS (2) | BODY •

form-data | x-www-form-urlencoded | raw • | binary | JSON (application/json)

□ Перенос строк

```
{
    "id":"{{tag.ALERT_ID}}",
    "reason":"responded"
}
```

□ Тестирование

Регулярное выражение для обработки результатов

Тайм-аут для выполнения команды (в секундах): 10

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. In the **Autostart criteria** settings block, 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. In the Incident **Response Actions** settings block, click **Add** → **Run connector**. In the opened window, select the connector that should be run when the rule is triggered:

    a. In the **Connector** drop-down list select previously created connector.

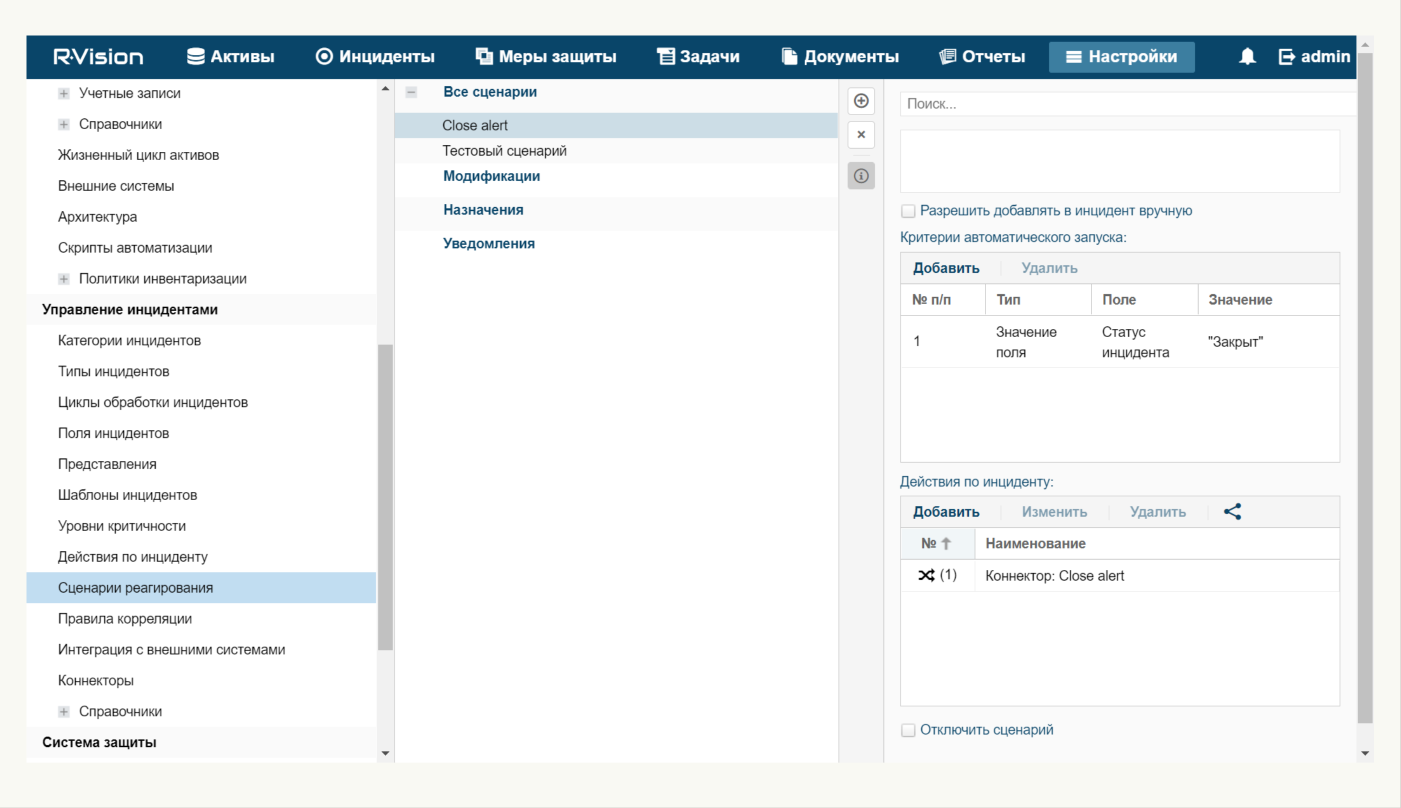
    b. Click **Add**.

    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.

R-Vision IRP version 4.0 playbook rule



□ Разрешить добавлять в инцидент вручную

Критерии автоматического запуска:

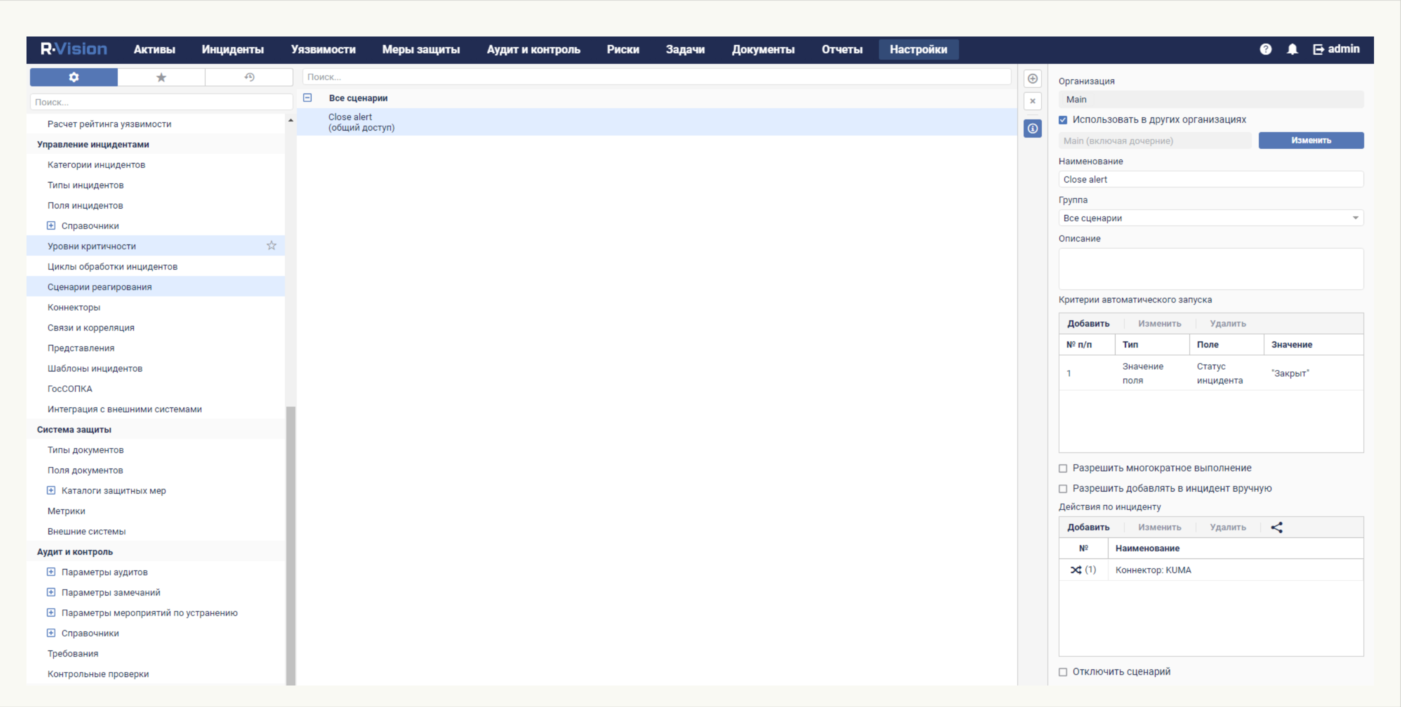
№ п/п	Тип	Поле	Значение
1	Значение поля	Статус инцидента	"Закрыт"

Действия по инциденту:

№	Наименование
(1)	Коннектор: Close alert

□ Отключить сценарий

R-Vision SOAR version 5.0 playbook rule



Организация

Main

☑ Использовать в других организациях

Main (включая дочерние)

Наименование

Close alert

Группа

Все сценарии

Описание

Критерии автоматического запуска

№ п/п	Тип	Поле	Значение
1	Значение поля	Статус инцидента	"Закрыт"

□ Разрешить многократное выполнение

□ Разрешить добавлять в инцидент вручную

Действия по инциденту

№	Наименование
(1)	Коннектор: KUMA

□ Отключить сценарий

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 application 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 under **Settings** → **LDAP server** in 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`
- `co`
- `company`
- `department`
- `description`
- `displayName`
- `distinguishedName`
- `division`
- `employeeID`
- `givenName`
- `l`
- `lastLogon`
- `lastLogonTimestamp`
- `Mail`
- `mailNickname`
- `managedObjects`
- `manager`
- `memberOf` (this attribute can be used for search during correlation)
- `mobile`
- `name`
- `objectCategory`
- `objectGUID` (this attribute always requested from Active Directory even if a user doesn't specify it)
- `objectSID`
- `physicalDeliveryOfficeName`

- `pwdLastSet`
- `sAMAccountName`
- `sAMAccountType`
- `sn`
- `streetAddress`
- `telephoneNumber`
- `title`
- `userAccountControl`
- `UserPrincipalName`
- `whenChanged`
- `whenCreated`

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 **Settings** → **LDAP server** and select the tenant for which you want to enable or disable all LDAP connections.

   The **LDAP server integration by tenant** window opens.

2. Select or clear the **Disabled** check box.

3. Click **Save**.

To enable or disable a specific LDAP connection:

1. In the KUMA web interface, open **Settings** → **LDAP server** and select the tenant for which you want to enable or disable an LDAP connection.

   The **LDAP server integration** window opens.

2. Select the relevant connection and either select or clear the **Disabled** check box in the opened window.

3. 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. Open the KUMA web interface and select **Settings** → **LDAP server**.

   The **LDAP server integration by tenant** window opens.

2. Click the **Add tenant** button.

   The **LDAP server integration** window is displayed.

3. In the **Tenant** drop-down list, select the tenant that you need to add.

4. Click **Save**.

The selected tenant is added to the LDAP server integration list.

To delete a tenant from the list of tenants for integration with an LDAP server:

1. Open the KUMA web interface and select **Settings** → **LDAP server**.

   The **LDAP server integration by tenant** window is displayed.

2. Select the check box next to the tenant that you need to delete, and click **Delete**.

3. Confirm deletion of the tenant.

The selected tenant is deleted from the LDAP server integration list.

Creating an LDAP server connection

To create a new LDAP connection to Active Directory:

1. In the KUMA web interface, open **Settings** → **LDAP server**.

2. Select or create a tenant for which you want to create a LDAP connection.

   The **LDAP server integration by tenant** window opens.

3. Click the **Add connection** button.

   The **Connection parameters** window opens.

4. Add a secret containing the account credentials for connecting to the Active Directory server. To do so:

   a. If you previously added a secret, in the **Secret** drop-down list, select the existing secret (with the **credentials** type).

      The selected secret can be changed by clicking on the ✎ button.

   b. If you want to create a new secret, click the + button.

      The **Secret** window opens.

   c. In the **Name** (required) field, enter the name of the secret containing 1 to 128 Unicode characters.

   d. In the **User** and **Password** (required) fields, enter the account credentials for connecting to the Active Directory server.

      You can enter the user name in one of the following formats: <user name>@<domain> or <domain><user name>.

   e. In the **Description** field, enter a description of up to 4,000 Unicode characters.

   f. Click the **Save** button.

5. In the **Name** (required) field, enter the unique name of the LDAP connection.

   The length of the string must be 1 to 128 Unicode characters.

6. In the **URL** (required) field, enter the address of the domain controller in the format `<hostname or IP address of server>:<port>`.

   > In case of server availability issues, you can specify multiple servers with domain controllers by separating them with commas. All of the specified servers must reside in the same domain.

7. If you want to use TLS encryption for the connection with the domain controller, select one of the following options from the **Type** drop-down list:

   - **startTLS**.

     When the startTLS method is used, first it establishes an unencrypted connection over port 389, then it sends an encryption request. If the STARTTLS command ends with an error, the connection is terminated.

     > Make sure that port 389 is open. Otherwise, a connection with the domain controller will be impossible.

   - **ssl**.

     When using SSL, an encrypted connection is immediately established over port 636.

   - **insecure**.

   When using an encrypted connection, it is impossible to specify an IP address as a URL.

8. If you enabled TLS encryption at the previous step, add a TLS certificate. To do so:

   a. If you previously uploaded a certificate, select it from the **Certificate** drop-down list.

      If no certificate was previously added, the drop-down list shows **No data**.

   b. If you want to upload a new certificate, click the + button on the right of the **Certificate** list.

      The **Secret** window opens.

   c. In the **Name** field, enter the name that will be displayed in the list of certificates after the certificate is added.

   d. Click the **Upload certificate file** button to add the file containing the Active Directory certificate. X.509 certificate public keys in Base64 are supported.

   e. If necessary, provide any relevant information about the certificate in the **Description** field.

   f. Click the **Save** button.

   The certificate will be uploaded and displayed in the **Certificate** list.

9. In the **Timeout in seconds** field, indicate the amount of time to wait for a response from the domain controller server.

If multiple addresses are indicated in the **URL** field, KUMA will wait the specified number of seconds for a response from the first server. If no response is received during that time, the program will contact the next server, and so on. If none of the indicated servers responds during the specified amount of time, the connection will be terminated with an error.

10. In the **Base DN** field, enter the base distinguished name of the directory in which you need to run the search query.

11. In the **Custom AD Account Attributes** field, specify the additional attributes that you want to use to enrich events ?.

Before configuring event enrichment using custom attributes, make sure that custom attributes are configured in AD.

To enrich events with accounts using custom attributes:

1. Add **Custom AD Account Attributes** in the LDAP connection settings.

   Standard **imported attributes from AD** cannot be added as custom attributes. For example, if you add the standard `accountExpires` attribute as a custom attribute, KUMA returns an error when saving the connection settings.

The following account attributes can be requested from Active Directory:

- `accountExpires`
- `badPasswordTime`
- `cn`
- `co`
- `company`
- `department`
- `description`
- `displayName`
- `distinguishedName`
- `division`
- `employeeID`
- `givenName`
- `l`
- `lastLogon`
- `lastLogonTimestamp`
- `Mail`
- `mailNickname`
- `managedObjects`
- `manager`
- `memberOf` (this attribute can be used for search during correlation)
- `mobile`
- `name`
- `objectCategory`
- `objectGUID` (this attribute always requested from Active Directory even if a user doesn't specify it)
- `objectSID`
- `physicalDeliveryOfficeName`

- `pwdLastSet`
- `sAMAccountName`
- `sAMAccountType`
- `sn`
- `streetAddress`
- `telephoneNumber`
- `title`
- `userAccountControl`
- `UserPrincipalName`
- `whenChanged`
- `whenCreated`

After you add custom attributes in the LDAP connection settings, the **LDAP attribute to receive** drop-down list in the collector automatically includes the new attributes. Custom attributes are identified by a question mark next to the attribute name. If you added the same attribute for multiple domains, the attribute is listed only once in the drop-down list. You can view the domains by moving your cursor over the question mark. Domain names are displayed as links. If you click a link, the domain is automatically added to **LDAP accounts mapping** if it was not previously added.

If you deleted a custom attribute in the LDAP connection settings, manually delete the row containing the attribute from the mapping table in the collector. Account attribute information in KUMA is updated each time you import accounts.

2. Import accounts.
3. In the collector, in the **LDAP mapping** table, define the rules for mapping KUMA fields to LDAP attributes.
4. Restart the collector.

   After the collector is restarted, KUMA begins enriching events with accounts.

12. Select the **Disabled** check box if you do not want to use this LDAP connection.

    This check box is cleared by default.

13. Click the **Save** button.

The LDAP connection to Active Directory will be created and displayed in the **LDAP server integration** window.

Account information from Active Directory will be requested immediately after the connection is saved, and then it will be updated at the specified frequency.

If you want to use multiple LDAP connections simultaneously for one tenant, you need to make sure that the domain controller address indicated in each of these connections is unique. Otherwise KUMA lets you enable only one of these connections. When checking the domain controller address, the program does not check whether the port is unique.

Creating a copy of an LDAP server connection

You can create an LDAP connection by copying an existing connection. In this case, all settings of the original connection are duplicated in the newly created connection.

To copy an LDAP connection:

1. In the KUMA web interface, open **Settings** → **LDAP server** and select the tenant for which you want to copy an LDAP connection.

   The **LDAP server integration** window opens.

2. Select the relevant connection.

3. In the opened **Connection parameters** window, click the **Duplicate connection** button.

   The New Connection window opens. The word `copy` will be added to the connection name.

4. If necessary, change the relevant settings.

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 program does not check whether the port is unique.

Changing an LDAP server connection

To change an LDAP server connection:

1. Open the KUMA web interface and select **Settings** → **LDAP server**.

   The **LDAP server integration by tenant** window opens.

2. Select the tenant for which you want to change the LDAP server connection.

   The **LDAP server integration** window opens.

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 is changed. Restart the KUMA services that use LDAP server data enrichment for the changes to take effect.

Changing the data update frequency

KUMA queries the LDAP server to update account data. This occurs:

- Immediately after creating a new connection.
- Immediately after changing the settings of an existing connection.
- According to a regular schedule every several hours. Every 12 hours by default.
- Whenever a user creates a task to update account data.

When querying LDAP servers, a task is created in the **Task manager** section of the KUMA web interface.

To change the schedule of KUMA queries to LDAP servers:

1. In the KUMA web interface, open **Settings** → **LDAP server** → **LDAP server integration by tenant**.
2. Select the relevant tenant.

   The **LDAP server integration** window opens.
3. In the **Data refresh interval** field, specify the required frequency in hours. The default value is 12.

The query schedule has been changed.

Changing the data storage period

Received user account data is stored in KUMA for 90 days by default if information about these accounts is no longer received from the Active Directory server. After this period, the data is deleted.

After KUMA account data is deleted, new and existing events are no longer enriched with this information. Account information will also be unavailable in alerts. If you want to view information about accounts throughout the entire period of alert storage, you must set the account data storage period to be longer than the alert storage period.

To change the storage period for the account data:

1. In the KUMA web interface, open **Settings** → **LDAP server** → **LDAP server integration by tenant**.
2. Select the relevant 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 **Settings** → **LDAP server** → **LDAP server integration by tenant**.
2. Select the relevant tenant.

   The **LDAP server integration** window opens.
3. Click the **Import accounts** button.

A task to receive account data from the selected tenant is added to the **Task manager** section of the KUMA web interface.

To start an account data update task for one LDAP connection of a tenant:

1. In the KUMA web interface, open **Settings** → **LDAP server** → **LDAP server integration by tenant**.
2. Select the relevant tenant.

   The **LDAP server integration** window opens.
3. Select the relevant LDAP server connection.

   The **Connection parameters** window opens.
4. Click the **Import accounts** button.

A task to receive account data from the selected connection of the tenant is added to the **Task manager** section of the KUMA web interface.

Deleting an LDAP server connection

To delete LDAP connection to Active Directory:

1. In the KUMA web interface, open **Settings** → **LDAP server** and select the tenant that owns the relevant LDAP connection.

   The **LDAP server integration** window opens.
2. Click the LDAP connection that you want to delete and click the **Delete** button.

3. 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 if it is disabled.**

   Domain authorization is enabled by default, but a connection to the domain 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 application 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 program 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 least privileges will be used. If multiple groups are matched for different tenants, the user will be assigned the specified role in each tenant.

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 authorization is disabled in the KUMA settings.
- An error occurred when entering the group of roles.
- The domain user name contains a space.

Enabling and disabling domain authentication

Domain authorization is enabled by default, but a connection to the domain is not configured. 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 authorization of users in the KUMA web interface:

1. In the application web interface, select **Settings** → **Domain authorization**.
2. In the **Authorization type** drop-down list, select one of the options:
   - FreeIPA
   - AD/ADFS
3. Do one of the following:
   - To disable domain authentication, select the **Disabled** check box in the upper part of the workspace.
   - To enable domain authentication, clear the **Disabled** check box in the upper part of the workspace.
4. Click the **Save** button.

   The selected settings are 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 application web interface, select **Settings** → **Domain authorization**.
2. In the **Authorization type** drop-down list, select **FreeIPA**.
3. In the **FreeIPA** settings block, 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.
4. 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.

5. 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.

   - **ssl**.

     When using SSL, an encrypted connection is immediately established over port 636.

   - **insecure**.

   When using an encrypted connection, it is impossible to specify an IP address as a URL.

6. 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 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.

7. 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 indicated in the **URL** field, KUMA waits for the specified number of seconds for a response from the first server. If no response is received during that time, the application contacts the next server. If none of the indicated servers responds during the specified amount of time, the connection will be terminated with an error.

8. 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.

9. 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.

   > If the user belongs to several groups within the same tenant, the role with the least privileges is used.

   Filter input example: `CN=KUMA team,OU=Groups,OU=Clients,DC=test,DC=domain`.

10. 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 application web interface, select **Settings** → **Domain authorization**.

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 **Test**.

   After clicking the **Test** 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 application web interface, select **Settings** → **Domain authorization**.

2. Under **Administration groups**, click **Add role groups**.

3. In the **Tenant** drop-down list, select the tenant of the users for whom you want to configure domain authentication.

4. In the **Selected 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**

   After you select the roles, a group filter field is displayed for each role. In the fields for each role, specify the DistinguishedName of the domain group. The users of this domain group must have the capability to perform authentication with their domain accounts. Group input example: `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** → **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, Access to 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. 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** → **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 the **Save** button.

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 under **Settings** → **Users**. 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 application web interface, select **Settings** → **Domain authorization**.

2. In the **Authorization type** drop-down list, select **AD/ADFS**.

3. In the **Active Directory** group of settings, in the **Base DN** field, enter the DistinguishedName of the root record to search for access groups in the Active Directory catalog service.

4. 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.

5. 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.

   - **ssl**.

  When using SSL, an encrypted connection is immediately established over port 636.

- **insecure**.

When using an encrypted connection, it is impossible to specify an IP address as a URL.

6. 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.

7. In the **Timeout in seconds** field, indicate the amount of time to wait for a response from the domain controller server.

   If multiple addresses are indicated in the **URL** field, KUMA waits for the specified number of seconds for a response from the first server. If no response is received during that time, the application contacts the next server. If none of the indicated servers responds during the specified amount of time, the connection will be terminated with an error.

8. 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.

   > If the user belongs to several groups within the same tenant, the role with the least privileges is used.

   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 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 application web interface, select **Settings** → **Domain authorization**.

2. In the **Authorization type** drop-down list, select **AD/ADFS**.

3. In the **Test connection** 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.

   The following formats for specifying a user are available in the **User** field: UserPrincipalName and domain\user.

4. Click **Test**.

   After clicking the **Test** 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 application web interface, select **Settings** → **Domain authorization**.

2. Under **Administration groups**, click **Add role groups**.

3. In the **Tenant** drop-down list, select the tenant of the users for whom you want to configure domain authentication.

4. In the **Selected 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**

   After you select the roles, a group filter field is displayed for each role. In the fields for each role, specify the DistinguishedName of the domain group. The users of this domain group must have the capability to perform authentication with their domain accounts. Group input example: `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** → **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, Access to 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. 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** → **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 the **Save** button.

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 under **Settings** → **Users**. 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 ADFS 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 ADFS are not specified or the **Disabled** check box is selected for ADFS 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 ADFS domain controller:

1. In the application web interface, select **Settings** → **Domain authorization**.
2. In the **Authorization type** drop-down list, select **AD/ADFS**.
3. In the **Active Directory Federation Services** settings block, in the **Client ID** field, enter the KUMA ID from the **Client ID** field in the ADFS.
4. In the **Relying party identifier** field, enter the KUMA ID from the **Relying party identifiers** field in the ADFS.
5. Enter the **Connect Metadata URI** from the **Connect Metadata URI** field. This parameter consists of the host where the ADFS resides (https://adfs.example.com), and the endpoint setting (/adfs/.well-known/openid-configuration).

   For example, https://adfs.example.com/adfs/.well-known/openid-configuration).
6. Enter the **ADFS redirect URL** from the **Redirect URL** field in the ADFS. The value of the **Redirect URL** field in the ADFS is defined when the Application group is configured. In the ADFS, you must 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/
7. 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.

   > If the user belongs to several groups within the same tenant, the role with the least privileges is used.

   Filter input example: `CN=KUMA team,OU=Groups,OU=Clients,DC=test,DC=domain`.

8. Click the **Save** button.

   A connection with the Active Directory Federation Services domain controller is now configured.

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 application web interface, select **Settings** → **Domain authorization**.
2. Under **Administration groups**, click **Add role groups**.
3. In the **Tenant** drop-down list, select the tenant of the users for whom you want to configure domain authentication.
4. In the **Selected 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**

   After you select the roles, a group filter field is displayed for each role. In the fields for each role, specify the DistinguishedName of the domain group. The users of this domain group must have the capability to perform authentication with their domain accounts. Group input example: `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** → **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, Access to 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. 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** → **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 the **Save** button.

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 under **Settings** → **Users**. 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 (ADFS).

The ADFS 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, the **Name** and Client Identifier

   fields are filled in automatically.

   Specify the value of the **Client Identifier** field in KUMA, when configuring domain authentication.

   In the

   Redirect URI field, enter the URI for redirection from ADFS 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 the **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**.

A new group is added. 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 ADFS, select the **Application groups** section and select the required connection group from the list. 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, open 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. Go to the **new-application-group - Web API Properties** window, open 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 ADFS is completed. You can proceed to configure domain authentication in KUMA.

Troubleshooting the Access denied error

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 is changed. To fix the error and resume domain authentication, update the certificate thumbprint saved in KUMA.

To update the certificate thumbprint on an Astra Linux or Oracle Linux host:

1. Contact Technical support to obtain the adfs_fingerprint_changer_tool binary file.

2. Place the received adfs_fingerprint_changer_tool binary file in any folder on the host with the KUMA Core. For example, /root/kuma-ansible-installer.

3. On the host with the KUMA Core, start the command line interpreter and use the `cd` command to go to the folder containing the adfs_fingerprint_changer_tool file.

   For example, you can enter the following command and press **Enter**:

   ```
   cd /root/kuma-ansible-installer
   ```

4. To grant the permissions to run a binary file and run the binary file, sequentially execute the following commands:

   ```
   chmod +x adfs_fingerprint_changer_tool
   ```

   ```
   ./adfs_fingerprint_changer_tool
   ```

To update the certificate thumbprint on a Kubernetes host:

1. Contact Technical support to obtain the adfs_fingerprint_changer_tool binary file.

2. Place the received adfs_fingerprint_changer_tool binary file in any folder on the computer of an administrator with access to the Kubernetes cluster and execute the following commands:

   ```
   k0s kubectl cp <path to adfs_fingerprint_changer_tool> $(k0s kubectl get pod -l
   app=core -n kuma -o name | cut -d/ -f2):/tmp/ -c mongodb -n kuma
   ```

   ```
   k0s kubectl exec $(k0s kubectl get pod -l app=core -n kuma -o name) -c mongodb -n kuma
   -- bash -c "chmod a+x /tmp/adfs_fingerprint_changer_tool &&
   /tmp/adfs_fingerprint_changer_tool"
   ```

After you run the binary file, the certificate thumbprint is updated and the domain authentication by means of ADFS is again available.

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.

Data in KUMA and NCIRCC is synchronized every 5-10 minutes.

To create a connection to NCIRCC:

1. In the KUMA web interface, open **Settings** → **NCIRCC**.
2. In the **URL** field, enter the URL for accessing NCIRCC.
3. In the **Token** settings block, create or select an existing secret with the API token that was issued to your organization for a connection 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 the + button and specify the following settings:
        - **Name** (required)—unique name of the service 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. Click **Save**.

     The secret containing the token for connecting to NCIRCC will be created. It is saved under **Resources** → **Secrets** and is owned by the main tenant.

   The selected secret can be changed by clicking on the ✎ button.
4. In the **Affected system function** drop-down list, select the area of activity of your organization.

   Available company business sectors ?

- 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

5. In the **Company** field, indicate the name of your company. This data will be forwarded to NCIRCC when incidents are exported.

6. Use the **Location** drop-down list to specify where your company is located. This data will be forwarded to NCIRCC when incidents are exported.

7. If necessary, under **Proxy**, create or select an existing proxy server that must be used when connecting to NCIRCC.

8. 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, install and trust the certificate of the intermediate certification authority to the KUMA Core server:

1. Click the https://support.globalsign.com/ca-certificates/intermediate-certificates/alphassl-intermediate-certificates link, find the "AlphaSSL SHA256 G4 Intermediate Certificate", and click "View as BASE64".

2. Paste the displayed certificate strings into a file and add the file with the certificate strings as the secret in KUMA.

3. After installing the certificate, restart the Core server.

As a result, the certificate is installed and you can proceed with configuring the integration.

Integration with Security Vision Incident Response Platform

Security Vision Incident Response Platform (hereinafter referred to as Security Vision IRP) 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.

Security Vision IRP can be integrated with KUMA. After configuring integration in Security Vision IRP, you can perform the following tasks:

- Request information about alerts from KUMA. In Security Vision IRP, *incidents* are created based on received data.

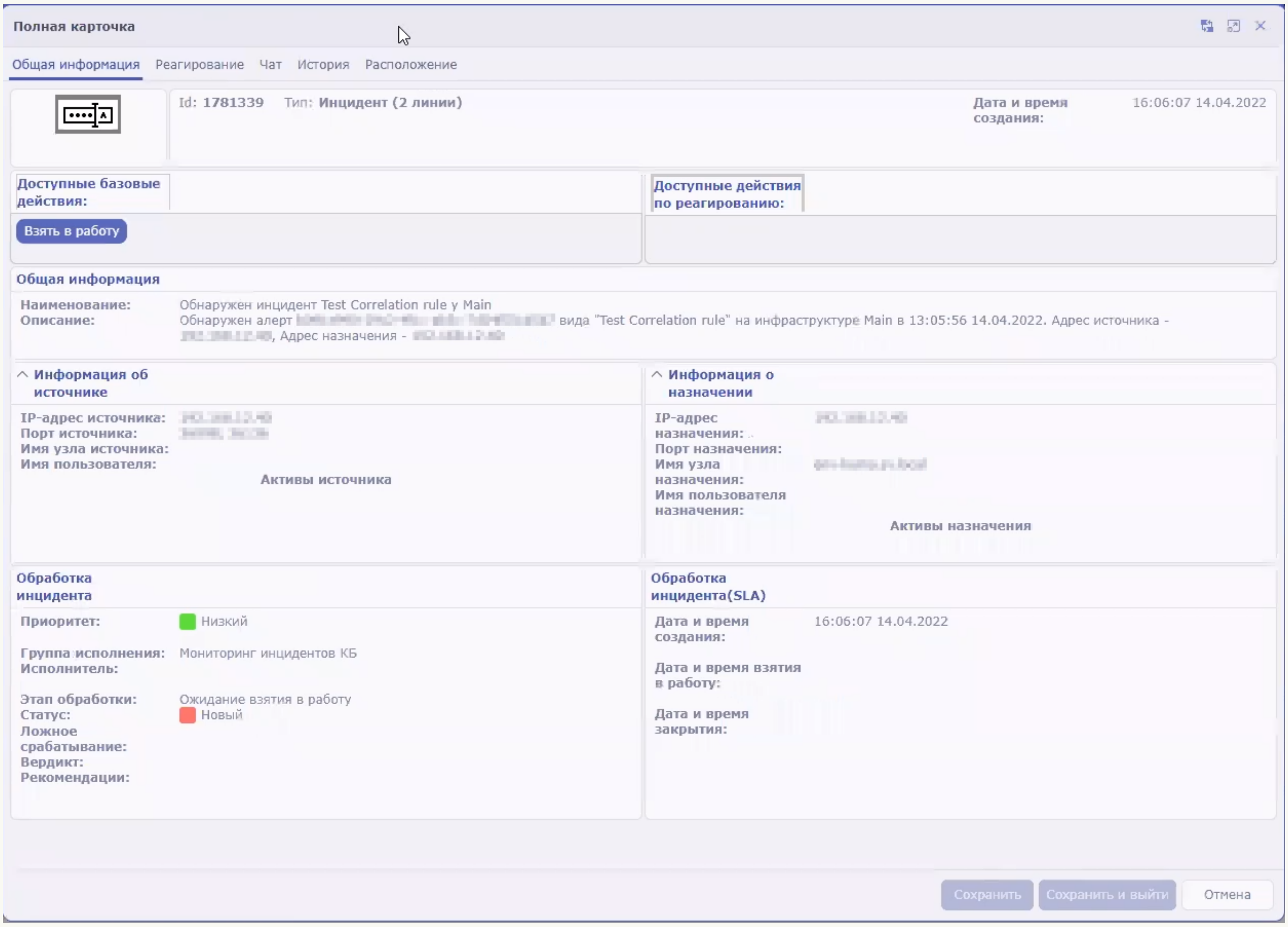
- Send requests to KUMA to close alerts.

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 Security Vision IRP vendor to learn more about the methods and conditions for obtaining a **Kaspersky KUMA** connector.

Working with Security Vision IRP incidents

Security Vision IRP incidents generated from KUMA alert data can be viewed in Security Vision IRP under **Incidents** → **Incidents (2 lines)** → **All incidents (2 lines)**. Events related to KUMA alerts are logged in each Security Vision IRP incident. Imported events can be viewed on the **Response** tab.

KUMA alert imported as Security Vision IRP incident



Id: 1781339 Тип: Инцидент (2 линии)

Дата и время создания: 16:06:07 14.04.2022

Доступные базовые действия:

Доступные действия по реагированию:

Общая информация

Наименование: Обнаружен инцидент Test Correlation rule у Main

Описание: Обнаружен алерт [redacted] вида "Test Correlation rule" на инфраструктуре Main в 13:05:56 14.04.2022. Адрес источника - [redacted], Адрес назначения - [redacted]

Информация об источнике

IP-адрес источника: [redacted]
Порт источника: [redacted]
Имя узла источника:
Имя пользователя:

Активы источника

Информация о назначении

IP-адрес назначения: [redacted]
Порт назначения:
Имя узла назначения: [redacted]
Имя пользователя назначения:

Активы назначения

Обработка инцидента

Приоритет: 🟩 Низкий

Группа исполнения: Мониторинг инцидентов КБ
Исполнитель:

Этап обработки: Ожидание взятия в работу
Статус: 🟥 Новый
Ложное срабатывание:
Вердикт:
Рекомендации:

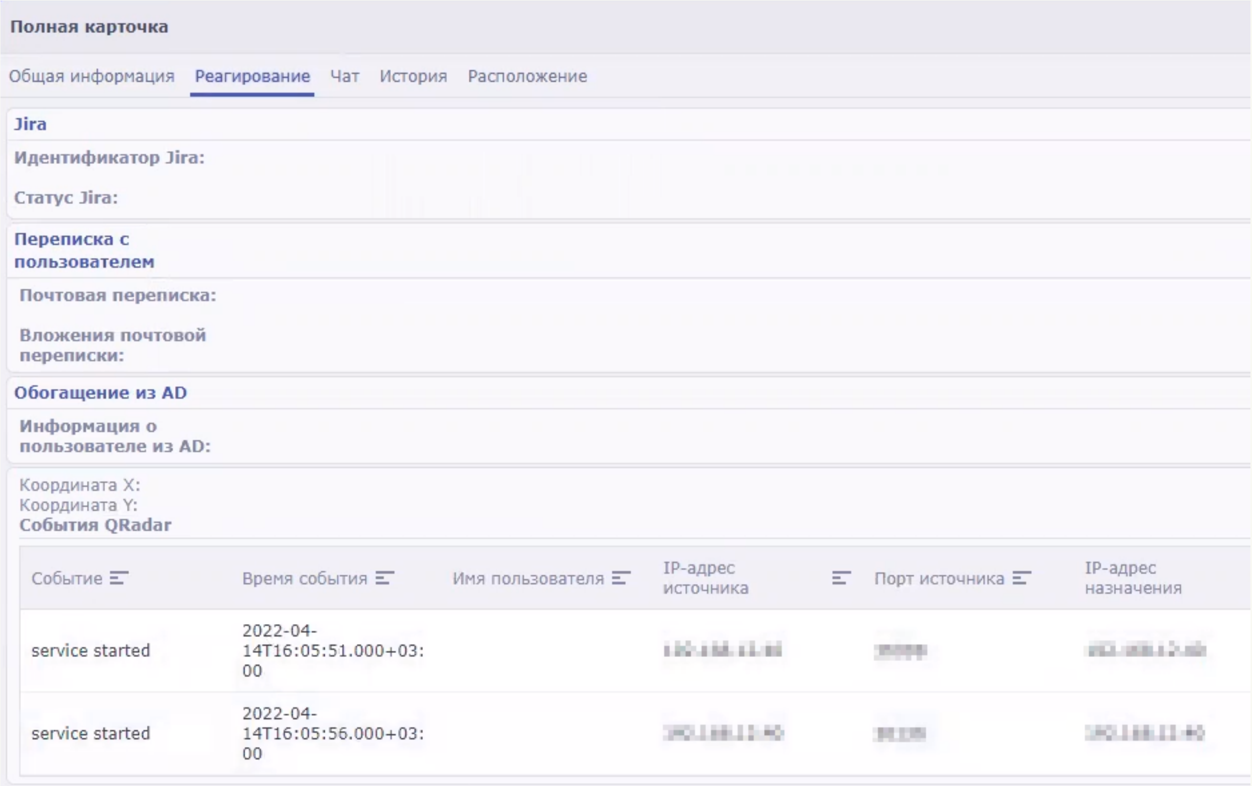
Обработка инцидента(SLA)

Дата и время создания: 16:06:07 14.04.2022

Дата и время взятия в работу:

Дата и время закрытия:

Security Vision IRP incident that was created based on KUMA alert



Jira

Идентификатор Jira:

Статус Jira:

Переписка с пользователем

Почтовая переписка:

Вложения почтовой переписки:

Обогащение из AD

Информация о пользователе из AD:

Координата X:
Координата Y:
События QRadar

Событие	Время события	Имя пользователя	IP-адрес источника	Порт источника	IP-адрес назначения
service started	2022-04-14T16:05:51.000+03:00		[redacted]	[redacted]	[redacted]
service started	2022-04-14T16:05:56.000+03:00		[redacted]	[redacted]	[redacted]

Events from KUMA alert that were imported to Security Vision IRP

Configuring integration in KUMA

To configure KUMA integration with Security Vision IRP, 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. This will be required to configure Security Vision IRP.

   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 indicated in the Security Vision IRP connector settings.

Configuring integration in Security Vision IRP

Configuration of integration in Security Vision IRP consists of importing and configuring a connector. If necessary, you can also change other Security Vision IRP settings related to KUMA data processing, such as the data processing schedule and worker.

For more detailed information about configuring Security Vision IRP, please refer to the product documentation.

Importing and configuring a connector

Adding a connector in Security Vision IRP

Integration of Security Vision IRP and KUMA is carried out by using the **Kaspersky KUMA** connector. Contact your Security Vision IRP vendor to learn more about the methods and conditions for obtaining a **Kaspersky KUMA** connector.

*To import a **Kaspersky KUMA** connector into Security Vision IRP:*

1. In Security Vision IRP, open **Settings** → **Connectors** → **Connectors**.

   You will see a list of connectors that have been added to Security Vision IRP.

2. At the top of the screen, click the import button and select the ZIP archive containing the **Kaspersky KUMA** connector.

The connector has been imported into Security Vision IRP 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 Security Vision IRP using the **Kaspersky KUMA** connector:*

1. In Security Vision IRP, open **Settings** → **Connectors** → **Connectors**.

   You will see a list of connectors that have been added to your Security Vision IRP.

2. Select the **Kaspersky KUMA** connector.

   The general settings of the connector will be displayed.

3. Under **Connector settings**, click the **Edit** button.

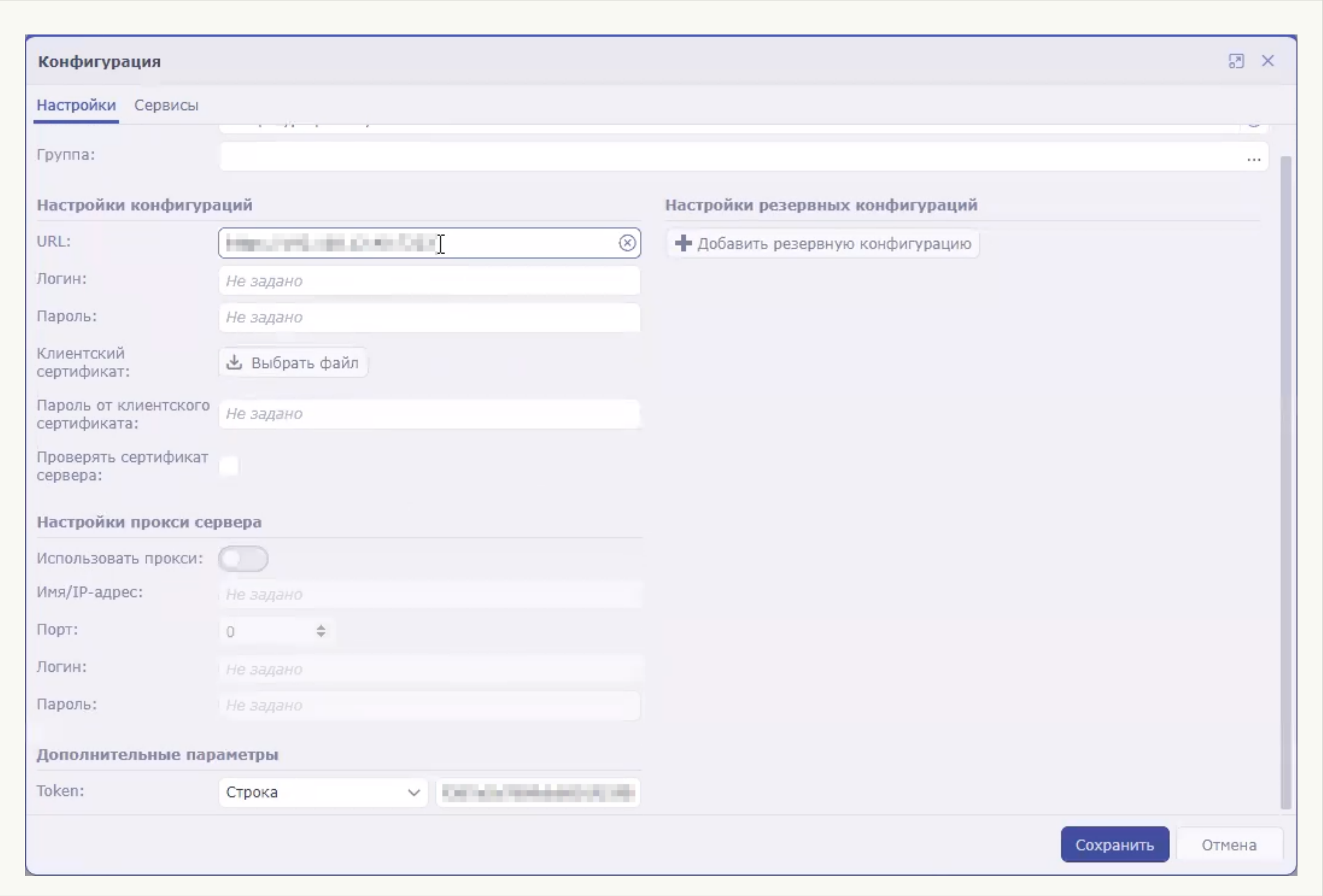
   The connector configuration will be displayed.

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 has been configured in the Security Vision IRP connector.

Security Vision IRP connector settings ?



Конфигурация

Настройки | Сервисы

Группа:

Настройки конфигураций

URL: [redacted]

Логин: *Не задано*

Пароль: *Не задано*

Клиентский сертификат: Выбрать файл

Пароль от клиентского сертификата: *Не задано*

Проверять сертификат сервера: □

Настройки прокси сервера

Использовать прокси:

Имя/IP-адрес: *Не задано*

Порт: 0

Логин: *Не задано*

Пароль: *Не задано*

Дополнительные параметры

Token: Строка [redacted]

Настройки резервных конфигураций

Configuring commands for interaction with KUMA in the Security Vision IRP connector

You can use Security Vision IRP to receive information about KUMA alerts (referred to as *incidents* in Security Vision IRP terminology) and send requests to close these alerts. To perform these actions, you need to configure the appropriate commands in the Security Vision IRP 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 Security Vision IRP 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 Security Vision IRP, open **Settings** → **Connectors** → **Connectors**.

   You will see a list of connectors that have been added to Security Vision IRP.

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 Security Vision IRP connector will query KUMA for information about all alerts with the **New** status and all events related to those alerts. The received data will be relayed to the Security Vision IRP handler, which will create Security Vision IRP incidents based on this data. If an already imported alert is updated in KUMA with additional information, new data will be imported to Security Vision IRP incident.

To configure a command to close KUMA alerts:

1. In Security Vision IRP, open **Settings** → **Connectors** → **Connectors**.

   You will see a list of connectors that have been added to Security Vision IRP.

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.

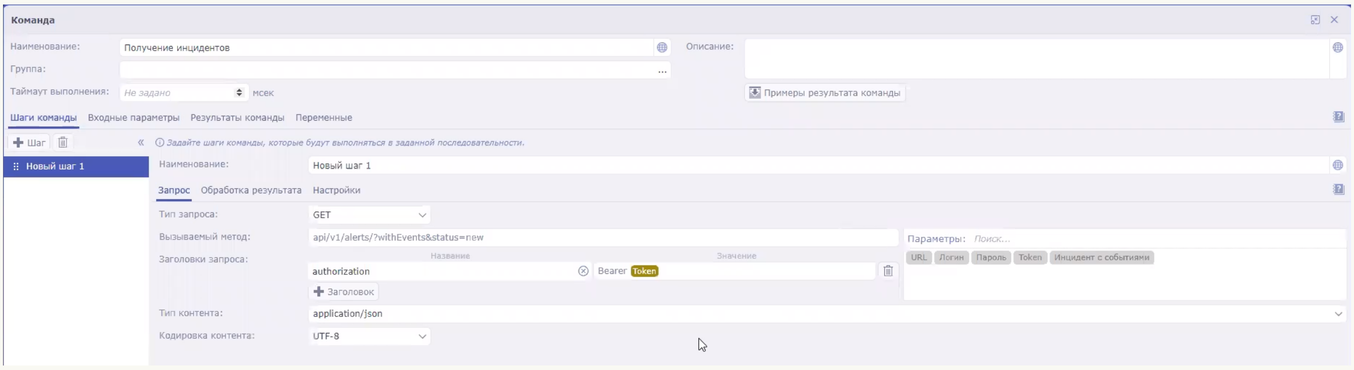
   - Under **Request headers**, in the **Name** field, indicate `authorization`. In the **Value** field, indicate **Bearer <token>**.

   - In the **Content type** drop-down list, select **application/json**.

5. Save the command and close the window.

The connector command is configured. When this command is executed, the incident will be closed in Security Vision IRP and the corresponding alert will be closed in KUMA.

Creating commands in Security Vision IRP



Наименование: Получение инцидентов

Группа:

Таймаут выполнения: Не задано мсек

Описание:

Задайте шаги команды, которые будут выполняться в заданной последовательности.

Новый шаг 1

Наименование: Новый шаг 1

Тип запроса: GET

Вызываемый метод: api/v1/alerts/?withEvents&status=new

Параметры: Поиск...

Заголовки запроса:

Название	Значение
authorization	Bearer Token

Тип контента: application/json

Кодировка контента: UTF-8

After configuring the connector, KUMA alerts will be sent to the platform as Security Vision IRP incidents. Then you need to configure incident handling in Security Vision IRP based on the security policies of your organization.

Configuring the handler, schedule, and worker process

Security Vision IRP handler

The Security Vision IRP handler receives KUMA alert data from the Security Vision IRP connector and creates Security Vision IRP incidents based on this data. A predefined **KUMA (Incidents)** handler is used for processing data. The settings of the **KUMA (Incidents)** handler are available in Security Vision IRP 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 Security Vision IRP under **Settings** → **Event processing** → **Schedule**:

- In the **Connector settings** block, you can configure the settings for starting the connector.
- In the **Handler settings** block, you can configure the settings for starting the handler.

Security Vision IRP worker process

The life cycle of Security Vision IRP incidents created based on KUMA alerts follows the preconfigured **Incident processing (2 lines)** worker. The worker can be configured in Security Vision IRP under **Settings** → **Workers** → **Worker templates**: select the **Incident processing (2 lines)** worker and click the transaction or state that you need to change.

Kaspersky Industrial CyberSecurity for Networks integration

Kaspersky Industrial CyberSecurity for Networks (hereinafter referred to as "KICS for Networks") is an application designed to protect the industrial enterprise infrastructure from information security threats, and to ensure uninterrupted operation. The application analyzes industrial network traffic to identify deviations in the values of process parameters, detect signs of network attacks, and monitor the operation and current state of network devices.

KICS for Networks version 4.0 or later can be integrated with KUMA. After configuring integration, you can perform the following tasks in KUMA:

- Import asset information from KICS for Networks to KUMA.
- Send asset status change commands from KUMA to KICS for Networks.

> Unlike KUMA, KICS for Networks refers to assets as devices.

The integration of KICS for Networks and KUMA must be configured in both applications:

1. In KICS for Networks, you need to create a KUMA connector and save the communication data package of this connector.
2. In KUMA, the communication data package of the connector is used to create a connection to KICS for Networks.

> The integration described in this section applies to importing asset information. KICS for Networks can also be configured to send events to KUMA. To do so, you need to create a SIEM/Syslog connector in KICS for Networks, and configure a collector on the KUMA side.

Configuring integration in KICS for Networks

> The program supports integration with KICS for Networks version 4.0 or later.

> It is recommended to configure integration of KICS for Networks and KUMA after ending Process Control rules learning mode. For more details, please refer to the documentation on KICS for Networks.

On the KICS for Networks side, integration configuration consists of creating a *KUMA-type connector*. In KICS for Networks, connectors are specialized application modules that enable KICS for Networks to exchange data with recipient systems, including KUMA. For more details on creating connectors, please refer to the documentation on KICS for Networks.

When a connector is added to KICS for Networks, a *communication data package* is automatically created for this connector. This is an encrypted configuration file for connecting to KICS for Networks that is used when configuring integration on the KUMA side.

Configuring integration in KUMA

It is recommended to configure integration of KICS for Networks and KUMA after ending Process Control rules learning mode. For more details, please refer to the documentation on KICS for Networks.

To configure integration with KICS for Networks in KUMA:

1. Open the KUMA web interface and select **Settings** → **Kaspersky Industrial CyberSecurity for Networks**.

   The **Kaspersky Industrial CyberSecurity for Networks integration by tenant** window opens.

2. Select or create a tenant for which you want to create an integration with KICS for Networks.

   The **Kaspersky Industrial CyberSecurity for Networks integration** window opens.

3. Click the **Communication data package** field and select the communication data package that was created in KICS for Networks.

4. In the **Communication data package password** field, enter the password of the communication data package.

5. Select the **Enable response** check box if you want to change the statuses of KICS for Networks assets by using KUMA response rules.

6. Click **Save**.

Integration with KICS for Networks is configured in KUMA, and the window shows the IP address of the node where the KICS for Networks connector will be running and its ID.

Enabling and disabling integration with KICS for Networks

To enable or disable KICS for Networks integration for a tenant:

1. In the KUMA web interface, open **Settings** → **Kaspersky Industrial CyberSecurity for Networks** and select the tenant for which you want to enable or disable KICS for Networks integration.

   The **Kaspersky Industrial CyberSecurity for Networks integration** window opens.

2. Select or clear the **Disabled** check box.

3. Click **Save**.

Changing the data update frequency

KUMA queries KICS for Networks to update its asset information. This occurs:

- Immediately after creating a new integration.
- Immediately after changing the settings of an existing integration.
- According to a regular schedule every several hours. This occurs every 3 hours by default.
- Whenever a user creates a task for updating asset data.

When querying KICS for Networks, a task is created in the **Task manager** section of the KUMA web interface.

To edit the schedule for importing information about KICS for Networks assets:

1. In the KUMA web interface, open **Settings** → **Kaspersky Industrial CyberSecurity for Networks**.
2. Select the relevant tenant.

   The **Kaspersky Industrial CyberSecurity for Networks integration** window opens.
3. In the **Data refresh interval** field, specify the required frequency in hours. The default value is 3.

The import schedule has been changed.

Special considerations when importing asset information from KICS for Networks

Importing assets

Assets are imported according to the asset import rules. Only assets with the **Authorized** and **Unauthorized** statuses are imported.

KICS for Networks assets are identified by a combination of the following parameters:

- IP address of the KICS for Networks instance with which the integration is configured.
- KICS for Networks connector ID is used to configure the integration.
- ID assigned to the asset (or "device") in the KICS for Networks instance.

Importing vulnerability information

When importing assets, KUMA also receives information about active vulnerabilities in KICS for Networks. If a vulnerability has been flagged as Remediated or Negligible in KICS for Networks, the information about this vulnerability is deleted from KUMA during the next import.

Information about asset vulnerabilities is displayed in the localization language of KICS for Networks in the **Asset details** window in the **Vulnerabilities** settings block.

> In KICS for Networks, vulnerabilities are referred to as risks and are divided into several types. All types of risks are imported into KUMA.

Imported data storage period

If information about a previously imported asset is no longer received from KICS for Networks, the asset is deleted after 30 days.

Changing the status of a KICS for Networks asset

After configuring integration, you can change the statuses of KICS for Networks assets from KUMA. Statuses can be changed either automatically or manually.

Asset statuses can be changed only if you enabled a response in the settings for connecting to KICS for Networks.

Manually changing the status of a KICS for Networks asset

Users with the General administrator, Tenant administrator, and Tier 2 analyst roles can manually change the statuses of assets imported from KICS for Networks in the tenants available to them.

To manually change a KICS for Networks 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 for Networks** drop-down list, select the status that you need to assign to the KICS for Networks asset. The *Authorized* or *Unauthorized* statuses are available.

The asset status is changed. The new status is displayed in KICS for Networks and in KUMA.

Automatically changing the status of a KICS for Networks asset

Automatic changes to the statuses of KICS for Networks assets are implemented using response rules. The rules must be added to the correlator, which will determine the conditions for triggering these rules.

Integration with Neurodat SIEM IM

Neurodat SIEM IM is an information security monitoring system.

You can configure the export of KUMA events to Neurodat SIEM IM. Based on incoming events and correlation rules, Neurodat SIEM IM automatically generates information security incidents.

To configure integration with Neurodat SIEM IM:

1. Connect to the Neurodat SIEM IM server over SSH using an account with administrative privileges.

2. Create a backup copy of the /opt/apache-tomcat-<server version>/conf/neurodat/soz_settings.properties configuration file.

3. In the /opt/apache-tomcat-<server version>/conf/neurodat/soz_settings.properties configuration file, edit the following settings as follows:

   - `kuma.on=true`

     This setting is an attribute of Neurodat SIEM IM interaction with KUMA.

   - `job_kuma=com.cbi.soz.server.utils.scheduler.KumaIncidentsJob`

   - `jobDelay_kuma=5000`

   - `jobPeriod_kuma=60000`

4. Save changes of the configuration file.

5. Run the following command to restart the tomcat service:

   ```
   sudo systemctl restart tomcat
   ```

6. Obtain a token for the user in KUMA. To do so:

   a. Open the KUMA web interface, click the name of your user account in the bottom-left corner of the window and click the **Profile** button in the opened menu.

      This opens the **User** window with your user account settings.

   b. Click the **Generate token** button.

      The **New token** window opens.

   c. If necessary, set the token expiration date:

      - Select the **No expiration date** check box.

      - In the **Expiration date** field, use the calendar to specify the date and time when the created token will expire.

   d. Click the **Generate token** button.

      The **Token** field with an automatically generated token is displayed in the user details area. Copy it.

      When the window is closed, the token is no longer displayed. If you did not copy the token before closing the window, you will have to generate a new token.

   e. Click **Save**.

7. Log in to Neurodat SIEM IM using the 'admin' account or another account that has the Administrator role for the organization you are configuring or the Administrator role for all organizations.

8. In the **Administration** → **Organization structure** menu item, select or create an organization that you want to receive incidents from KUMA.

9. On the organization form, do the following:

   a. Select the **Configure integration with KUMA** check box.

   b. In the **KUMA IP address and port** field, specify the KUMA API address, for example, `https://192.168.58.27:7223/api/v1/`.

   c. In the **KUMA API key** field, specify the user token obtained at step 6.

   d. Save the organization information.

Integration with KUMA is configured.

Neurodat SIEM IM tests access to KUMA and, if successful, displays a message about being ready to receive data from KUMA.

Kaspersky Automated Security Awareness Platform

Kaspersky Automated Security Awareness Platform (hereinafter also referred to as "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. Open the KUMA web interface and select **Settings** → **Kaspersky Automated Security Awareness Platform**.

   The **Kaspersky Automated Security Awareness Platform** window opens.
2. In the **Secret** field click the + button to create a secret of the **token** by entering the token received from ASAP:
   a. In the **Name** field, enter the name of the secret. Must contain 1 to 128 Unicode characters.

   b. In the **Token** field, enter the authorization token for API requests to ASAP.

   c. If necessary, add the secret description in the **Description** field.

   d. Click **Save**.
3. In the **ASAP Open API URL** field, specify the address used by ASAP for API requests.
4. 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.
5. If necessary, in the **Proxy** drop-down list select the proxy server resource to be used to connect to ASAP.
6. To disable or enable integration with ASAP, select or clear the **Disabled** check box.
7. 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 the name of the rule.

   c. In the **Tenant** drop-down list, select the tenant that owns the resource.

   d. In the **Type** drop-down list, select **Run script**.

   e. In the **Script name** field, enter the name of the script.

   f. In the **Script arguments** field, enter `{{.Name}}`.
   This passes the name of the correlation event as the argument of the script.

   g. Click **Save**.

2. Add the response rule to the correlator:

   a. In the **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 the name of the rule.

   c. In the **Tenant** drop-down list, select the tenant that owns the resource.

   d. In the **Type** drop-down list, select **Run script**.

   e. In the **Script name** field, enter the name of the script. `ug.py`.

   f. In the **Script arguments** field, specify:

      - one of the operations depending on the type of the object being blocked:

        - `blockurl` to block access by URL
        - `blockip` to block access by IP address
        - `blockdomain` to block access by domain name

- `-i {{<KUMA field from which the value of the blocked object must be taken, depending on the operation>}}`

  Example:

  ```
  blockurl -i {{.RequetstUrl}}
  ```

g. In the **Conditions** section, add conditions corresponding to correlation rules that require blocking in UserGate when triggered.

h. Click **Save**.

2. Add the response rule to the correlator:

a. In the **Resources** → **Correlators** section, select the correlator that must respond and in whose directory you placed the script.

b. In the steps tree, select **Response rules**.

c. Click **Add**.

d. In the **Response rule** drop-down list, select the rule added at step 1 of these instructions.

e. In the steps tree, select **Setup validation**.

f. Click **Save and reload services**.

g. Click the **Save** button.

The response rule is linked to the correlator and ready to use.

Integration with Kaspersky Web Traffic Security

This integration is an example and may require additional configuration depending on the versions used and the specifics of the infrastructure.
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 **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. Download the script and library:

   https://box.kaspersky.com/d/2dfd1d677c7547a7ac1e/

3. Place the downloaded script on the KUMA correlator server at the following path: /opt/kaspersky/kuma/correlator/`<correlator ID from step 1>`/scripts/.

4. Connect to the correlator server via SSH and go to the path from step 3:

```
cd /opt/kaspersky/kuma/correlator/< correlator ID from step 1 >/scripts/
```

5. Run the following command:

   ```
   chmod +x kwts.py kwtsWebApiV6.py && chown kuma:kuma kwts.py kwtsWebApiV6.py
   ```

The script is ready to use.

Configuring a response rule for integration with KWTS

To configure a response rule:

1. Create a response rule:

   a. In the KUMA 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 the name of the rule.

   c. In the **Tenant** drop-down list, select the tenant that owns the resource.

   d. In the **Type** drop-down list, select **Run script**.

   e. In the **Script name** field, enter the name of the script, kwts.py.

   f. In the **Script arguments** field, specify:

   - `--host` — address of the KWTS server.
   - `--username` — name of the user account created in KWTS or local administrator.
   - `--password` — KWTS user account password.
   - `--rule_id` — ID of the rule created in KWTS.
   - Specify one of the options depending on the type of the object being blocked:
     - `--url` — specify the field of the KUMA event from which you want to obtain the URL, for example, `{{.RequestUrl}}`.
     - `--ip` — specify the field of the KUMA event from which you want to obtain the IP address, for example, `{{.DestinationAddress}}`.
     - `--domain` — specify the field of the KUMA event from which you want to obtain the domain name, for example, `{{.DestinationHostName}}`.
   - `--ntlm` — specify this option if the KWTS user was created with NTLM authentication.

     Example:

     ```
     --host <address> --username <user> --password <pass> --rule_id <id> --url
     {{.RequestUrl}}
     ```

   g. In the **Conditions** section, add conditions corresponding to correlation rules that require blocking in KWTS when triggered.

   h. Click **Save**.

2. Add the response rule to the correlator:

   a. In the **Resources** → **Correlators** section, select the correlator that must respond and in whose directory you placed the script.

   b. In the steps tree, select **Response rules**.

   c. Click **Add**.

   d. In the **Response rule** drop-down list, select the rule added at step 1 of these instructions.

   e. In the steps tree, select **Setup validation**.

   f. Click **Save and reload services**.

   g. Click the **Save** button.

   The response rule is linked to the correlator and ready to use.

Integration with Kaspersky Secure Mail Gateway

> This integration is an example and may require additional configuration depending on the versions used and the specifics of the infrastructure.
> 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. Download the script and library:

    https://box.kaspersky.com/d/2dfd1d677c7547a7ac1e/

3. Place the downloaded script on the KUMA correlator server at the following path: /opt/kaspersky/kuma/correlator/`<correlator ID from step 1>`/scripts/.

4. Connect to the correlator server via SSH and go to the path from step 3:

    ```
    cd /opt/kaspersky/kuma/correlator/< correlator ID from step 1 >/scripts/
    ```

5. Run the following command:

    ```
    chmod +x ksmg.py ksmgWebApiV2.py && chown kuma:kuma ksmg.py ksmgWebApiV2.py
    ```

The script is ready to use.

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 the name of the rule.

   c. In the **Tenant** drop-down list, select the tenant that owns the resource.

   d. In the **Type** drop-down list, select **Run script**.

   e. In the **Script name** field, enter the name of the script, 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.

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 example.kuma.com:7223 --token 949fc03d97bad5d04b6e231c68be54fb
   --tenant Main --vuln-report /home/user/vuln.csv --inventory-report
   /home/user/inventory.csv
   ```

   You can use additional flags and commands for import operations. For example, the `-v` command displays an extended report on the received assets. A detailed description of the available flags and commands is provided in the "Flags and commands of redcheck-tool.py" table. You can also use the `--help` command to view information on the available flags and commands.

The asset information is imported from the RedCheck report to KUMA. The console displays information on the number of new and updated assets.

Example:

```
inventory has been imported for 2 host(s)
software has been imported for 5 host(s)
vulnerabilities has been imported for 4 host(s)
```

Example of extended import information:

```
[inventory import] Host: localhost Code: 200 Response: {'insertedIDs': {'0':
'52ca11c6-a0e6-4dfd-8ef9-bf58189340f8'}, 'updatedCount': 0, 'errors': []}
[inventory import] Host: 10.0.0.2 Code: 200 Response: {'insertedIDs': {'0':
'1583e552-5137-4164-92e0-01e60fb6edb0'}, 'updatedCount': 0, 'errors': []}
[software import][error] Host: localhost Skipped asset with FQDN localhost or IP
127.0.0.1
[software import] Host: 10.0.0.2 Code: 200 Response: {'insertedIDs': {},
'updatedCount': 1, 'errors': []}
[vulnerabilities import] Host: 10.0.0.2 Code: 200 Response: {'insertedIDs': {},
'updatedCount': 1, 'errors': []}
[vulnerabilities import] Host: 10.0.0.1 Code: 200 Response: {'insertedIDs': {'0':
'0628f683-c20c-4107-abf3-d837b3dbbf01'}, 'updatedCount': 0, 'errors': []}
[vulnerabilities import] Host: localhost Code: 200 Response: {'insertedIDs': {},
'updatedCount': 1, 'errors': []}
[vulnerabilities import] Host: 10.0.0.3 Code: 200 Response: {'insertedIDs': {'0':
'ed01e0a8-dcb0-4609-ab2b-91e50092555d'}, 'updatedCount': 0, 'errors': []}
inventory has been imported for 2 host(s)
software has been imported for 1 host(s)
```

```
vulnerabilities has been imported for 4 host(s)
```

The tool works as follows when importing assets:

- KUMA overwrites the data of assets imported through the API, and deletes information about their resolved vulnerabilities.
- KUMA skips assets with invalid data.

Flags and commands of redcheck-tool.py

Flags and commands	Mandatory	Description
`--kuma-rest <address and port of the KUMA server>`	Yes	Port 7223 is used for API requests by default. You can change the port if necessary.
`--token <token>`	Yes	The value of the option must contain only the token. The 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 cfqdn=localhost or ip=127.0.0.1 was skipped.

Configuring receipt of Sendmail events

You can configure the receipt of Sendmail mail agent events in the KUMA SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring Sendmail logging.
2. Configuring the event source server.

3. Creating a KUMA collector.

   To receive Sendmail events, use the following values in the Collector Installation Wizard:

   - At the **Event parsing** step, select the **[OOTB] Sendmail syslog** normalizer.
   - At the **Transport** step, select the **tcp** or **udp** connector type.

4. Installing KUMA collector.

5. Verifying receipt of Sendmail events in the KUMA collector

   You can verify that the Sendmail event source server is correctly configured in the Searching for related events section of 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 administrative privileges.

2. In the /etc/rsyslog.d/ directory, create the Sendmail-to-siem.conf file and add the following line to it:

   ```
   If $programname contains 'sendmail' then @<<IP address of the collector>:<port of the collector>>
   ```

   Example:

   ```
   If $programname contains 'sendmail' then @192.168.1.5:1514
   ```

   If you want to send events via TCP, the contents of the file must be as follows:

   ```
   If $programname contains 'sendmail' then @@<<IP address of the collector>:<port of the collector>>
   ```

3. Create a backup copy of the /etc/rsyslog.conf file.

4. Add the following lines to the /etc/rsyslog.conf configuration file:

```
$IncludeConfig /etc/Sendmail-to-siem.conf
$RepeatedMsgReduction off
```

5. Save your changes.
6. Restart the rsyslog service by executing the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

Managing KUMA

This section describes the general settings of KUMA.

Logging in to the program web interface

To log in to the program 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.
4. Click the **Login** button.

The main window of the program web interface opens.

> In multitenancy mode, a user who is logging in to the program web interface for the first time will see the data only for those tenants that were selected for the user when their user account was created.

To log out of the program web interface,

open the KUMA web interface, click your user account name in the bottom-left corner of the window, and click the **Logout** button in the opened menu.

Viewing KUMA metrics

Comprehensive information about the performance of the KUMA Core, storage, collectors, and correlators is available in the **Metrics** section of the KUMA web interface. Selecting this section opens the Grafana portal deployed as part of KUMA Core installation and is updated automatically. If the **Metrics** section shows core: <port number>, this means that KUMA is deployed in a high availability configuration and the metrics were received from the host on which the **Core was installed**. In other configurations, the name of the host from which KUMA receives metrics is displayed.

To determine on which host the Core is running, run the following command in the terminal of one of the controllers:

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

The default Grafana user name and password are `admin` and `admin`.

Available metrics

Collector indicators:

- IO—metrics related to the service input and output.
  - Processing EPS—the number of processed events per second.
  - Processing Latency—the time required to process a single event (the median is displayed).
  - Output EPS—the number of events, sent to the destination per second.
  - Output Latency—the time required to send a batch of events to the destination and receive a response from it (the median is displayed).
  - Output Errors—the number or errors when sending event batches to the destination per second. Network errors and errors writing the disk buffer are displayed separately.
  - Output Event Loss—the number of lost events per second. Events can be lost due to network errors or errors writing the disk buffer. Events are also lost if the destination responded with an error code (for example, if the request was invalid).
- Normalization—metrics related to the normalizers.
  - Raw & Normalized event size—the size of the raw event and size of the normalized event (the median is displayed).
  - Errors—the number of normalization errors per second.
- Filtration—metrics related to the filters.
  - EPS—the number of events rejected by the Collector per second. The collector only rejects events if the user has added a filter into the collector service configuration.
- Aggregation—metrics related to the aggregation rules.
  - EPS—the number of events received and created 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 the enrichment rules.
  - Cache RPS—the number requests to the local cache per second.

    - Source RPS—the number of requests to the enrichment source (for example, the Dictionary resource).
    - Source Latency—the time required to send a request to the enrichment source and receive a response from it (the median is displayed).
    - Queue—the enrichment requests queue size. This metric helps to find bottleneck enrichment rules.
    - Errors—the number of enrichment source request errors per second.

Correlator metrics

- IO—metrics related to the service input and output.
    - Processing EPS—the number of processed events per second.
    - Processing Latency—the time required to process a single event (the median is displayed).
    - Output EPS—the number of events, sent to the destination per second.
    - Output Latency—the time required to send a batch of events to the destination and receive a response from it (the median is displayed).
    - Output Errors—the number or errors when sending event batches to the destination per second. Network errors and errors writing the disk buffer are displayed separately.
    - Output Event Loss—the number of lost events per second. Events can be lost due to network errors or errors writing the disk buffer. Events are also lost if the destination responded with an error code (for example, if the request was invalid).
- Correlation—metrics related to the correlation rules.
    - EPS—the number of correlation events created per second.
    - Buckets—the number of buckets in the correlation rule (only for the standard kind of correlation rules).
- Active Lists—metrics related to the active lists.
    - RPS—the number of requests (and their type) to the Active list per second.
    - Records—the number of entries in the Active list.
    - WAL Size—the size of the Write-Ahead-Log. This metric helps determine the size of the Active list.

Storage indicators

- IO—metrics related to the service input and output.
    - RPS—the number of requests to the Storage service per second.
    - Latency—the time of proxying a single request to the ClickHouse node (the median is displayed).

Core service metrics

- IO—metrics related to the service input and output.
    - RPS—the number of requests to the Core service per second.

    - Latency—the time of processing a single request (the median is displayed).
    - Errors—the number of request errors per second.
- Notification Feed—metrics related to user activity.
    - Subscriptions—the number of clients, connected to the Core via SSE to receive server messages in real time. This number usually correlates with the number of clients using the KUMA web interface.
    - Errors—the number of message sending errors per second.
- Schedulers—metrics related to Core tasks.
    - Active—the number of repeating active system tasks. The tasks created by the user are ignored.
    - Latency—the time of processing a single request (the median is displayed).
    - Position—the position (timestamp) of the alert creation task. The next ClickHouse scan for correlation events will start from this position.
    - Errors—the number of task errors per second.

General metrics common for all services

- Process—general process metrics.
    - CPU—CPU usage.
    - Memory—RAM usage (RSS).
    - DISK IOPS—the number of disk read/write operations per second.
    - DISK BPS—the number of bytes read/written to the disk per second.
    - Network BPS—the number of bytes received/sent per second.
    - Network Packet Loss—the number of network packets lost per second.
    - GC Latency—the time of the GO Garbage Collector cycle (the median is displayed).
    - Goroutines—the number of active goroutines. This number differs from the thread count.
- OS—metrics related to the operating system.
    - Load—the average load.
    - CPU—CPU usage.
    - Memory—RAM usage (RSS).
    - Disk—disk space usage.

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:
   1. systemctl daemon-reload
   2. systemctl restart kuma-victoria-metrics

The storage period for metrics has been changed.

Managing KUMA tasks

When working in the program 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.

Viewing the tasks table

The tasks table contains a list of created tasks and is located in the **Task manager** section of the program web interface window.

You can view the tasks that were created by you (current user). A user with the General Administrator role can view the tasks of all users.

By default, the **Display only my own** filter is applied in the **Task manager** section. To see tasks, clear the check box from the **Display only my own** filter.

The tasks table contains the following information:

- **State**—the state of the task. One of the following statuses can be assigned to a task:
  - *Green dot blinking*—the task is active.
  - **Completed**—the task is complete.
  - **Cancel**—the task was canceled by the user.
  - **Error**—the task was not completed because of an error. The error message is displayed if you hover the mouse over the exclamation mark icon.
- **Task**—the task type. The program provides the following types of tasks:
  - **Events export**—export KUMA events.
  - **Threat Lookup**—request data from the Kaspersky Threat Intelligence Portal.
  - **Retroscan**—task for replaying events.

    - **KSC assets import**—imports asset data from Kaspersky Security Center servers.
    - **Accounts import**—imports user data from Active Directory.
    - **KICS for Networks assets import**—imports asset data from KICS for Networks.
    - **Repository update**—updates the KUMA repository to receive the resource packages from the source specified in settings.
- **Created by**—the user who created the task. If the task was created automatically, the column will show **Scheduled task**.
- **Created**—task creation time.
- **Updated**—time when the task was last updated.
- **Tenant**—the name of the tenant in which the task was started.

The task date format depends on the localization language selected in the application settings. Possible date format options:

- English localization: YYYY-MM-DD.
- Russian localization: DD.MM.YYYY.

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 and order of columns in the tasks table:

1. In the KUMA web interface, select the **Task manager** section.

   The tasks table is displayed.
2. In the table header, click the ⚙ button.
3. In the opened window, do the following:
    - 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. Click the link containing the task type in the **Task** column.

   A list of the operations available for this task type will be displayed.

3. Select **Show results**.

The task results window opens.

> In this section, the **Display only my own** filter is applied by default in the **Created by** column of the task table. To view all tasks, disable this filter.

Restarting a task

To restart a task:

1. In the KUMA web interface, select the **Task manager** section.

   The tasks table is displayed.

2. Click the link containing the task type in the **Task** column.

   A list of the operations available for this task type will be displayed.

3. Select **Restart**.

The task will be restarted.

Proxies

Proxy servers are used to store proxy server configuration settings, for example, in destinations. The http type is supported.

Available settings:

- **Name** (required)—unique name of the proxy server. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Use URL from the secret** (required)—drop-down list to select a secret resource that stores URLs of proxy servers. If required, a secret can be created in the proxy server creation window by using the + button. The selected secret can be changed by clicking on the ✎ button.

- **Do not use for domains**—one or more domains that require direct access.
- **Description**—up to 4,000 Unicode characters.

Connecting to an SMTP server

KUMA can be configured to send email notifications using an SMTP server. Users will receive notifications if the **Receive email notifications** check box is selected in their profile settings.

Only one SMTP server can be added to process KUMA notifications. An SMTP server connection is managed in the KUMA web interface under **Settings** → **General** → **SMTP server settings**.

To configure SMTP server connection:

1. Open the KUMA web interface and select **Settings** → **General**.

2. In the **SMTP server settings** block, change the relevant settings:

   - **Disabled**—select this check box if you want to disable connection to the SMTP server.
   - **Host** (required)—SMTP host in one of the following formats: hostname, IPv4, IPv6.
   - **Port** (required)—SMTP port. The value must be an integer from 1 to 65,535.
   - **From** (required)—email address of the message sender. For example, `kuma@company.com`.
   - **Alias for KUMA Core server**—name of the KUMA Core server that is used in your network. Must be different from the FQDN.
   - If necessary, use the **Secret** drop-down list to select a secret of the **credentials** type that contains the account credentials for connecting to the SMTP server.

     Add secret

     1. If you previously created a secret, select it from the **Secret** drop-down list.

        If no secret was previously added, the drop-down list shows **No data**.

     2. If you want to add a new secret, click the + button on the right of the **Secret** list.

        The **Secret** window opens.

     3. In the **Name** field, enter the name that will be used to display the secret in the list of available secrets.

     4. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector.

     5. If necessary, add any other information about the secret in the **Description** field.

     6. Click the **Save** button.

     The secret will be added and displayed in the **Secret** list.

- Select the necessary frequency of notifications in the **Monitoring notifications interval** drop-down list.

  Notifications from the source about a monitoring policy triggering are repeated after the selected period until the status of the source becomes green again.

  If the **Notify once** setting is selected, you receive a notification about monitoring policy activation only once.

- Turn on the **Disable monitoring notifications** toggle button if you do not want to receive notifications about the state of event sources. The toggle switch is turned off by default.

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 *Virus scan* task to start when Kaspersky Security Center registers the *Malicious object detected* event, specify this name in the **Value** field.

      You can view the event name in the **Name** field of the event details.

5. On the **Actions** tab, 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 processes that the service can run simultaneously.

   By default, the number of work processes is the same as the number of virtual processors on the server where the correlator service is installed.

- In the **Description** field, you can add up to 4,000 Unicode characters.

- In the **Filter** drop-down list, select the filter added at Step 3 of this instruction.

To send requests to Kaspersky Security Center, you must ensure that Kaspersky Security Center is available over the UDP protocol.

If a response rule is owned by the shared tenant, the displayed Kaspersky Security Center tasks that are available for selection are from the Kaspersky Security Center server that the main tenant is connected to.

If a response rule has a selected task that is absent from the Kaspersky Security Center server that the tenant is connected to, the task is not performed for assets of this tenant. This situation could arise when two tenants are using a common correlator, for example.

Step 5. Adding a response rule to the correlator

To add a response rule to the correlator:

1. In the KUMA 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 logs

KUMA provides the following types of logs:

- Installer logs
- Component logs

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 log is appended until it reaches 5 GB. When the log reaches 5 GB, it is archived and new events are written to a new log. Archives are kept in the log folder for 7 days, after 7 days the archive is deleted. A maximum of four archived logs are stored on the server at the same time. Whenever a new log archive is created, if the total number of archives becomes greater than four, the oldest log archive is deleted.

Debug mode is available for the following components:

Core	To enable it: in the KUMA web interface, select **Settings** → **General** → **Core settings** → **Debug**. Storage location: • /opt/kaspersky/kuma/core/log/stdout.log — the standard output stream of the service is redirected here. • /opt/kaspersky/kuma/core/log/stderr.log — the standard error stream of the service is redirected here. 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 name>/log/<service name>. You can download the service logs from the KUMA web interface, in the **Resources** → **Active services** section by selecting the desired service and clicking **Log**. Logs residing on Linux machines can be viewed by running the journalctl and tail command. For example: • Storage. To return the latest logs from the storage installed on the server, run the following command: `journalctl -f -u kuma-storage-< storage ID >` • Correlators. To return the latest logs from correlators installed on the server, run the following command:

	`journalctl -f -u kuma-correlator-< correlator ID >`
	• Collectors. To return the latest logs from a specific collector installed on the server, run the following command:
	`journalctl -f -u kuma-collector-< collector ID >`
	• Agents. To return the latest logs from an agent installed on the server, run the following command:
	`tail -f /opt/kaspersky/agent/< Agent ID >/log/agent`
	The activity of Agents on Windows machines is always logged if they are assigned the logon as a service permission. Data is specified in more detail when the **Debug** check box is selected. Agent logs on Windows machines can be viewed in the file located at the path %PROGRAMDATA%\Kaspersky Lab\KUMA\<Agent ID>\agent.log. Logs of Agents on Linux machines are stored in the agent installation directory.

Resources: • Connectors • Destinations • Enrichment rules	To enable it, use the **Debug** toggle switch in the settings of the service to which the resource is linked. The logs are stored on the machine hosting the installed service that uses the relevant resource. Detailed data for resources can be viewed in the log of the service linked to a resource.

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
```

KUMA notifications

Standard notifications

KUMA can be configured to send email notifications using an SMTP server. To do so, configure a connection to an SMTP server and select the **Receive email notifications** check box for users who should receive notifications.

KUMA automatically notifies users about the following events:

- A report was created (the users listed in the report template receive a notification).
- 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).

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.

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

Geographic data obtained from MaxMind and IP2Location can be used in KUMA if the data files are first converted to a format supported by KUMA. Conversion can be done using the script below. Make sure that the files do not contain duplicate records. For example, if a file has few columns, different records may contain data from the same network with the same geodata. Such files cannot be converted. To successfully perform the conversion, make sure that there are no duplicate rows and that every row has at least one unique field.

Download script

Python 2.7 or later is required to run the script.

Script start command:

```
python converter.py --type <type of geographic data being processed: "maxmind" or
"ip2location"> --out <directory where a CSV file containing geographic data in KUMA
format will be placed> --input <path to the ZIP archive containing geographic data from
MaxMind or IP2location>
```

When the script is run with the `--help` flag, help is displayed for the available script parameters: `python converter.py --help`

Command for converting a file containing a Russian database of IP address ranges from a MaxMind ZIP archive:

```
python converter.py --type maxmind --lang ru --input MaxMind.zip --out
geoip_maxmind_ru.csv
```

If the `--lang` parameter is not specified, the script receives information from the GeoLite2-City-Locations-en.csv file from the ZIP archive by default.

Absence of the `--lang` parameter for MaxMind is equivalent to the following command:

```
python converter.py --type maxmind --input MaxMind.zip --out geoip_maxmind.csv
```

Command for converting a file from an IP2Location ZIP archive:

```
python converter.py --type ip2location --input IP2LOCATION-LITE-DB11.CSV.ZIP --out
geoip_ip2location.csv
```

Command for converting a file from several IP2Location ZIP archives:

```
python converter.py --type ip2location --input IP2LOCATION-LITE-DB11.CSV.ZIP
IP2LOCATION-LITE-DB11.IPV6.CSV.ZIP --out geoip_ip2location_ipv4_ipv6.csv
```

The `--lang` parameter is not used for IP2Location.

Required sets of fields

The MaxMind source files GeoLite2-City-Blocks-IPv4.csv and GeoLite2-City-Blocks-IPv6.csv must contain the following set of fields:

network,geoname_id,registered_country_geoname_id,represented_country_geoname_id,
is_anonymous_proxy,is_satellite_provider,postal_code,latitude,longitude,accuracy_radius

Example set of source data:

```
network,geoname_id,registered_country_geoname_id,represented_country_geoname_id,
is_anonymous_proxy,is_satellite_provider,postal_code,latitude,longitude,accuracy_radius

1.0.0.0/24,2077456,2077456,,0,0,,-33.4940,143.2104,1000

1.0.1.0/24,1814991,1814991,,0,0,,34.7732,113.7220,1000
```

The remaining CSV files with the locale code must contain the following set of fields:

geoname_id,locale_code,continent_code,continent_name,country_iso_code,country_name,
subdivision_1_iso_code,subdivision_1_name,subdivision_2_iso_code,subdivision_2_name,
city_name,metro_code,time_zone,is_in_european_union

Example set of source data:

```
geoname_id,locale_code,continent_code,continent_name,country_iso_code,country_name,
subdivision_1_iso_code,subdivision_1_name,subdivision_2_iso_code,subdivision_2_name,
city_name,metro_code,time_zone,is_in_european_union

1392,de,AS,Asien,IR,Iran,02,Mazandaran,,,,,Asia/Tehran,0

7240,de,AS,Asien,IR,Iran,28,Nord-Chorasan,,,,,Asia/Tehran,0
```

The source IP2Location files must contain data on the network ranges, Country, Region, City, Latitude, and Longitude

Example set of source data:

```
"0","16777215","-","-","-","-","0.000000","0.000000","-","-"

"16777216","16777471","US","United States of America","California","Los
Angeles","34.052230","-118.243680","90001","-07:00"

"16777472","16778239","CN","China","Fujian","Fuzhou","26.061390","119.306110","350004","+0
```

If the source files contain a different set of fields than the one indicated in this section, or if some fields are missing, the missing fields in the target CSV file will be empty after conversion.

Importing and exporting geographic data

If necessary, you can manually import and export geographic data into KUMA. Geographic data is imported and exported in a CSV file. If the geographic data import is successful, the previously added data is overwritten and an audit event is generated in KUMA.

To import geographic data into KUMA:

1. Prepare a CSV file containing geographic data.

   Geographic data received from MaxMind and IP2Location must be converted to a format supported by KUMA.

2. In the KUMA web interface, open **Settings** → **General**.

3. In the **Geographic data** settings block, click the **Import from file** button and select a CSV file containing geographic data.

   Wait for the geographic data import to finish. The data import is interrupted if the page is refreshed.

The geographic data is uploaded to KUMA.

To export geographic data from KUMA:

1. In the KUMA web interface, open **Settings** → **General**.
2. In the **Geographic data** settings block, click the **Export** button.

Geographic data will be downloaded as a CSV file named geoip.csv (in UTF-8 encoding) based on the settings of your browser.

The data is exported in the same format as it was uploaded, with the exception of IP address ranges. If a range of addresses was indicated in the format `1.0.0.0/24` in a file imported into KUMA, the range will be displayed in the format `1.0.0.0-1.0.0.255` in the exported file.

Default mapping of geographic data

If you select the `SourceAddress`, `DestinationAddress` and `DeviceAddress` event fields as the IP address source when configuring a geographic data enrichment rule, the **Apply default mapping** button becomes available. You can 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 chapter 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 conditions for rejecting or selecting individual events from the stream of events.
- **Response rules**—resources of this type are used in correlators to, for example, execute scripts or launch Kaspersky Security Center tasks when certain conditions are met.
- **Notification templates**—resources of this type are used when sending notifications about new alerts.
- **Active lists**—resources of this type are used by correlators for dynamic data processing when analyzing events according to correlation rules.
- **Dictionaries**—resources of this type are used to store keys and their values, which may be required by other KUMA resources and services.
- **Proxies**—resources of this type contain settings for using proxy servers.
- **Secrets**—resources of this type are used to securely store confidential information (such as credentials) that KUMA needs to interact with external services.

When you click on a resource type, a window opens displaying a table with the available resources of this type. The resource table contains the following columns:

- **Name**—the name of a resource. Can be used to search for resources and sort them.
- **Updated**—the date and time of the last update of a resource. Can be used to sort resources.
- **Created by**—the name of the user who created a resource.
- **Description**—the description of a resource.

The maximum table size is not limited. If you want to select all resources, scroll to the end of the table and select the **Select all** check box, which selects all available resources in the table.

Resources can be organized into folders. The folder structure is displayed in the left part of the window: root folders correspond to tenants and contain a list of all resources of the tenant. All other folders nested within the root folder display the resources of an individual folder. When a folder is selected, the resources it contains are displayed as a table in the right pane of the window.

Resources can be created, edited, copied, moved from one folder to another, and deleted. Resources can also be exported and imported.

KUMA comes with a set of predefined resources, which can be identified by the "[OOTB]<resource_name>" name. OOTB resources are protected from editing.

If you want to adapt a predefined OOTB resource to your organization's infrastructure:

1. In the **Resources**-<resource type> section, select the OOTB resource that you want to edit.
2. In the upper part of the KUMA 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.

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 program does not delete folders that contain files or subfolders.

Creating, duplicating, moving, editing, and deleting resources

You can create, move, copy, edit, and delete resources.

To create the resource:

1. In the **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 unique resource name in the **Name** field.

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.

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**.

   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. Click **Save**.

The resource will be updated. If this resource is used in a service, restart the service to apply the new settings.

To delete the resource:

1. In the **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**.

   A confirmation window opens.
3. Click **OK**.

The resource will be deleted.

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 capability to use it without direct Internet access ensures 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. Specify the **Emails for notification** by clicking the **Add** button. The notifications that new packages or new versions of the packages imported into the tenant are available in the repository are sent to the specified email addresses.

   If you specify the email address of a KUMA user, the **Receive email notifications** check box must be selected in the user profile. For emails that do not belong to any KUMA user, the messages are received without additional settings. The settings for connecting to the SMTP server must be specified in all cases.

4. Click **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 with the KUMA Core

       If a local folder is used, the kuma user must have access to this folder and its contents.

3. Specify the **Emails for notification** by clicking the **Add** button. The notifications that new packages or new versions of the packages imported into the tenant are available in the repository are sent to the specified email addresses.

   If you specify the email address of a KUMA user, the **Receive email notifications** check box must be selected in the user profile. For emails that do not belong to any KUMA user, the messages are received without additional settings. The settings for connecting to the SMTP server must be specified in all cases.

4. Click **Run update**. Thus, you simultaneously save the settings and manually start the **Repository update** task.

Configuring a custom source using Kaspersky Update Utility

You can update resources without Internet access by using a custom update source via the Kaspersky Update Utility.

Configuration consists of the following steps:

1. Configuring a custom source using Kaspersky Update Utility:
   a. Installing and configuring Kaspersky Update Utility on one of the computers in the corporate LAN.
   b. Configuring copying of updates to a shared folder in Kaspersky Update Utility settings.
2. Configuring update of the KUMA repository from a custom source.

Configuring a custom source using Kaspersky Update Utility:

You can download the Kaspersky Update Utility distribution kit from the Kaspersky Technical Support website.

1. In Kaspersky Update Utility, enable the download of updates for KUMA 2.1:
   - Under **Applications** – **Perimeter control**, select the check box next to KUMA 2.1 to enable the update capability.
   - If you work with Kaspersky Update Utility using the command line, add the following line to the [ComponentSettings] section of the updater.ini configuration file or specify the `true` value for an existing line:

     ```
     KasperskyUnifiedMonitoringAndAnalysisPlatform_3_0=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.

5. Click the **Export** button.

The resources in a password-protected file are saved on your computer using your browser settings. The Secret resources are exported blank.

Importing resources

To import resources:

1. In the **Resources** section, click **Import resources**.

   The **Resource import** window opens.

2. In the **Tenant** drop-down list, select the tenant to assign the imported resources to.

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.

     The imported resources 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 resource copy. When importing future versions of the package, the duplicate is not updated because it is a separate object.

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 ensure the functioning of imported resources, you must create the corresponding fields of the extended event schema in a resource of the "normalizer" type.

If a normalizer that uses an extended event schema field is imported into KUMA and the same field already exists in KUMA, the previously created field is used.

About conflict resolving

When resources are imported into KUMA from a file, they are compared with existing resources; the following parameters are compared:

- Name and kind. If an imported resource's name and kind parameters match those of the existing one, the imported resource's name is automatically changed.
- ID. If identifiers of two resources match, a conflict appears that must be resolved by the user. This could happen when you import resources to the same KUMA server from which they were exported.

When resolving a conflict you can choose either to *replace existing resource* with the imported one or to *keep exiting resource*, skipping the imported one.

Some resources are linked: for example, in some types of connectors, the connector secret must be specified. The secrets are also imported if they are linked to a connector. Such linked resources are exported and imported together.

Special considerations of import:

1. Resources are imported to the selected tenant.
2. If a linked resource was in the Shared tenant, it ends up in the Shared tenant when imported.
3. In the **Conflicts** window, the **Parent** column always displays the top-most parent resource among those that were selected during import.
4. If a conflict occurs during import and you choose to replace existing resource with a new one, it would mean that all the other resources linked to the one being replaced are automatically replaced with the imported resources.

Known errors:

1. The linked resource ends up in the tenant specified during the import, and not in the Shared tenant, as indicated in the **Conflicts** window, under the following conditions:

   a. The associated resource is initially in the Shared tenant.

   b. In the **Conflicts** window, you select **Skip** for all parent objects of the linked resource from the Shared tenant.

   c. You leave the linked resource from the Shared tenant for replacement.

2. After importing, the categories do not have a tenant specified in the filter under the following conditions:

   a. The filter contains linked asset categories from different tenants.

   b. Asset category names are the same.

   c. You are importing this filter with linked asset categories to a new server.

3. In Tenant 1, the name of the asset category is duplicated under the following conditions:

   a. in Tenant 1, you have a filter with linked asset categories from Tenant 1 and the Shared tenant.

   b. The names of the linked asset categories are the same.

   c. You are importing such a filter from Tenant 1 to the Shared tenant.

4. You cannot import conflicting resources into the same tenant.

   The error "Unable to import conflicting resources into the same tenant" means that the imported package contains conflicting resources from different tenants and cannot be imported into the Shared tenant.

   Solution: Select a tenant other than Shared to import the package. In this case, during the import, resources originally located in the Shared tenant are imported into the Shared tenant, and resources from the other tenant are imported into the tenant selected during import.

5. Only the general administrator can import categories into the Shared tenant.

   The error "Only the general administrator can import categories into the Shared tenant" means that the imported package contains resources with linked shared asset categories. You can see the categories or resources with linked shared asset categories in the KUMA Core log. Path to the Core log:

   /opt/kaspersky/kuma/core/log/core

   Solution. Choose one of the following options:

   - Do not import resources to which shared categories are linked: clear the check boxes next to the relevant resources.

   - Perform the import under a General administrator account.

6. Only the general administrator can import resources into the Shared tenant.

   The error "Only the general administrator can import resources into the Shared tenant" means that the imported package contains resources with linked shared resources. You can see the resources with linked shared resources in the KUMA Core log. Path to the Core log:

   /opt/kaspersky/kuma/core/log/core

   Solution. Choose one of the following options:

   - Do not import resources that have linked resources from the Shared tenant, and the shared resources themselves: clear the check boxes next to the relevant resources.

   - Perform the import under a General administrator account.

Destinations

Destinations define network settings for sending normalized events. Collectors and correlators use destinations to describe where to send processed events. Typically, the destination points are the correlator and storage.

The settings of destinations are configured on two tabs: **Basic settings** and **Advanced settings**. The available settings depend on the selected type of destination:

- **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.
- **storage**—used to transmit data to the storage.
- **correlator**—used to transmit data to the correlator.

Destination, type nats-jetstream

The **nats-jetstream** type is used for NATS communications.

Basic settings tab

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
The **State** toggle switch	Used when events must be sent to the destination. By default, sending events is enabled.
Type	Required setting. Destination type, **nats-jetstream**.
URL	Required setting. URL that you want to connect to.
Topic	Required setting. The topic of NATS messages. Must contain Unicode characters.
Delimiter	Specify a character that defines where one event ends and the other begins. By default, \n is used.
Authorization	Type of authorization when connecting to the specified URL Possible values:

	- **disabled** is the default value. - **plain** — if this option is selected, you must indicate the secret containing user account credentials for authorization when connecting to the connector. Add secret 1. If you previously created a secret, select it from the **Secret** drop-down list. If no secret was previously added, the drop-down list shows **No data**. 2. If you want to add a new secret, click the + button on the right of the **Secret** list. The **Secret** window opens. 3. In the **Name** field, enter the name that will be used to display the secret in the list of available secrets. 4. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector. 5. If necessary, add any other information about the secret in the **Description** field. 6. Click the **Save** button. The secret will be added and displayed in the **Secret** list.
Description	Resource description: up to 4,000 Unicode characters.

Advanced settings tab

Setting	Description
Compression	You can use Snappy compression. By default, compression is **disabled**.
Buffer size	Sets the size of the buffer. The default value is 1 KB, and the maximum value is 64 MB.
Disk buffer size limit	Size of the disk buffer in bytes. The default value is 10 GB.
Cluster ID	ID of the NATS cluster.
Output format	Format for sending events to an external destination. Available values: - JSON - CEF
TLS mode	Use of TLS encryption. Available values: - **Disabled** (default) means TLS encryption is not used. - **Enabled** means encryption is used, but the certificate is not verified. - **With verification** means encryption is used with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created

automatically during program installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/.

- **Custom CA** means encryption is used with verification that the certificate was signed by a Certificate Authority. The secret containing the certificate is selected from the **Custom CA** drop-down list, which is displayed when this option is selected.
  Creating a certificate signed by a Certificate Authority

  > To use this TLS mode, you must do the following on the KUMA Core server (OpenSSL commands are used in the examples below):
  >
  > 1. Create the key that will be used by the Certificate Authority.
  >
  >   Example command:
  >
  >    ```
  >    openssl genrsa -out ca.key 2048
  >    ```
  >
  > 2. Generate a certificate for the key that was just created.
  >
  >   Example command:
  >
  >    ```
  >    openssl req -new -x509 -days 365 -key ca.key -subj "/CN=
  >    <common host name of Certificate Authority>" -out ca.crt
  >    ```
  >
  > 3. Create a private key and a request to have it signed by the Certificate Authority.
  >
  >   Example command:
  >
  >    ```
  >    openssl req -newkey rsa:2048 -nodes -keyout server.key -subj
  >    "/CN=<common host name of KUMA server>" -out server.csr
  >    ```
  >
  > 4. Create a certificate signed by the Certificate Authority. The subjectAltName must include the domain names or IP addresses of the server for which the certificate is being created.
  >
  >   Example command:
  >
  >    ```
  >    openssl x509 -req -extfile <(printf
  >    "subjectAltName=DNS:domain1.ru,DNS:domain2.com,IP:192.168.0.1")
  >    -days 365 -in server.csr -CA ca.crt -CAkey ca.key -
  >    CAcreateserial -out server.crt
  >    ```
  >
  > 5. The obtained `server.crt` certificate should be uploaded in the KUMA web interface as a **certificate**-type secret, which should then be selected from the **Custom CA** drop-down list.

When using TLS, it is impossible to specify an IP address as a URL.

Delimiter	In the drop-down list, you can select the character to mark the boundary between events. By default, `\n` is used.
Buffer flush interval	Time (in seconds) between sending batches of data to the destination. The default value is `1 second`.
Number of handlers	This field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
Debug	This toggle switch lets you specify whether resource logging must be enabled. The default value is **Disabled**.
Disk buffer disabled	Drop-down list that lets you enable or disable the disk buffer. By default, the disk buffer is enabled.

	The disk buffer is used if the collector cannot send normalized events to the destination. The amount of allocated disk space is limited by the value of the **Disk buffer size limit** setting. If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer.
Filter	In the **Filter** section, you can specify the criteria for identifying events that must be processed by the resource. You can select an existing filter from the drop-down list or **create** a new filter. Creating a filter in resources ?

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. The name must contain 1 to 128 Unicode characters.
4. In the **Conditions** settings block, specify the conditions that the events must meet:
   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

   Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

   Filter operators ?

- **=**—the left operand equals the right operand.

- **<**—the left operand is less than the right operand.

- **<=**—the left operand is less than or equal to the right operand.

- **>**—the left operand is greater than the right operand.

- **>=**—the left operand is greater than or equal to the right operand.

- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).

- **contains**—the left operand contains values of the right operand.

- **startsWith**—the left operand starts with one of the values of the right operand.

- **endsWith**—the left operand ends with one of the values of the right operand.

- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.

- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.

- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.

- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.

- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.

- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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.

Tcp type

The **tcp** type is used for TCP communications.

Basic settings tab

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
The **State** toggle switch	Used when events must be sent to the destination. By default, sending events is enabled.
Type	Required setting. Destination type, **tcp**.
URL	Required setting. URL that you want to connect to. Available formats: `host:port`, `IPv4:port`, `:port`.

	IPv6 addresses are also supported. When using IPv6 addresses, you must also specify the interface in the `[address%interface]:port` format. For example, `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`).
Description	Resource description: up to 4,000 Unicode characters.

Advanced settings tab

Setting	Description
Compression	You can use Snappy compression. By default, compression is **disabled**.
Buffer size	Sets the size of the buffer. The default value is 1 KB, and the maximum value is 64 MB.
Timeout	The time (in seconds) to wait for a response from another service or component. The default value is `30`.
Disk buffer size limit	Size of the disk buffer in bytes. The default value is 10 GB.
Output format	Format for sending events to an external destination. Available values: • JSON • CEF
TLS mode	TLS encryption mode using certificates in pem x509 format. Available values: • **Disabled** means TLS encryption is not used. The default value. • **Enabled** means encryption is used, but certificates are not verified. • **With verification** means encryption is used with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during application installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/. When using TLS, it is impossible to specify an IP address as a URL.
Delimiter	In the drop-down list, you can select the character to mark the boundary between events. By default, `\n` is used.
Buffer flush interval	Time (in seconds) between sending batches of data to the destination. The default value is `1 second`.
Number of handlers	This field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
Debug	This toggle switch lets you specify whether resource logging must be enabled. The default value is **Disabled**.
Disk buffer disabled	Drop-down list that lets you enable or disable the disk buffer. By default, the disk buffer is enabled. The disk buffer is used if the collector cannot send normalized events to the destination. The amount of allocated disk space is limited by the value of the **Disk buffer size limit** setting. If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer.
Filter	In this section, you can specify the criteria for identifying events that must be processed

by the resource. You can select an existing filter from the drop-down list or **create** a new filter.

Creating a filter in resources ?

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. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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.

Http type

The **http** type is used for HTTP communications.

Basic settings tab

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
The **State** toggle switch	Used when events must be sent to the destination. By default, sending events is enabled.
Type	Required setting. Destination type, **http**.
URL	Required setting. URL that you want to connect to. Available formats: `host:port`, `IPv4:port`, `:port`.

	IPv6 addresses are also supported, however, when you use them, you must specify the interface as well: `[address%interface]:port`. Example: `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`).
Authorization	Type of authorization when connecting to the specified URL Possible values: • **disabled** is the default value. • **plain:** if this option is selected, you must indicate the secret containing user account credentials for authorization when connecting to the connector. Add secret 1. If you previously created a secret, select it from the **Secret** drop-down list. If no secret was previously added, the drop-down list shows **No data**. 2. If you want to add a new secret, click the + button on the right of the **Secret** list. The **Secret** window opens. 3. In the **Name** field, enter the name that will be used to display the secret in the list of available secrets. 4. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector. 5. If necessary, add any other information about the secret in the **Description** field. 6. Click the **Save** button. The secret will be added and displayed in the **Secret** list.
Description	Resource description: up to 4,000 Unicode characters.

Advanced settings tab

Setting	Description
Compression	You can use Snappy compression. By default, compression is **disabled**.
Buffer size	Sets the size of the buffer. The default value is 1 KB, and the maximum value is 64 MB.
Timeout	The time (in seconds) to wait for a response from another service or component. The default value is `30`.
Disk buffer size limit	Size of the disk buffer in bytes. The default value is 10 GB.
Output format	Format for sending events to an external destination. Available values: • JSON • CEF
TLS mode	Use of TLS encryption. Available values:

- **Disabled** (default) means TLS encryption is not used.
- **Enabled** means encryption is used, but the certificate is not verified.
- **With verification** means encryption is used with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during program installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/.
- **Custom CA** means encryption is used with verification that the certificate was signed by a Certificate Authority. The secret containing the certificate is selected from the **Custom CA** drop-down list, which is displayed when this option is selected.

Creating a certificate signed by a Certificate Authority

To use this TLS mode, you must do the following on the KUMA Core server (OpenSSL commands are used in the examples below):

1. Create the key that will be used by the Certificate Authority.

   Example command:

   ```
   openssl genrsa -out ca.key 2048
   ```

2. Generate a certificate for the key that was just created.

   Example command:

   ```
   openssl req -new -x509 -days 365 -key ca.key -subj "/CN=
   <common host name of Certificate Authority>" -out ca.crt
   ```

3. Create a private key and a request to have it signed by the Certificate Authority.

   Example command:

   ```
   openssl req -newkey rsa:2048 -nodes -keyout server.key -subj
   "/CN=<common host name of KUMA server>" -out server.csr
   ```

4. Create a certificate signed by the Certificate Authority. The subjectAltName must include the domain names or IP addresses of the server for which the certificate is being created.

   Example command:

   ```
   openssl x509 -req -extfile <(printf
   "subjectAltName=DNS:domain1.ru,DNS:domain2.com,IP:192.168.0.1")
   -days 365 -in server.csr -CA ca.crt -CAkey ca.key -
   CAcreateserial -out server.crt
   ```

5. The obtained `server.crt` certificate should be uploaded in the KUMA web interface as a **certificate**-type secret, which should then be selected from the **Custom CA** drop-down list.

When using TLS, it is impossible to specify an IP address as a URL.

URL selection policy

From the drop-down list, you can select the method of deciding which URL to send events to if multiple URLs are specified. Available values:

- **Any**. Events are sent to one of the available URLs as long as this URL receives events. If the connection is broken (for example, the receiving node is disconnected) a different URL will be selected as the events destination.

	• **Prefer first**. Events are sent to the first URL in the list of added addresses. If it becomes unavailable, events are sent to the next available node in sequence. When the first URL becomes available again, events start to be sent to it again. • **Balanced** means that packages with events are evenly distributed among the available URLs from the list. Because packets are sent either on a destination buffer overflow or on the flush timer, this URL selection policy does not guarantee an equal distribution of events to destinations.
Delimiter	In the drop-down list, you can select the character to mark the boundary between events. By default, \n is used.
Path	The path that must be added for the URL request. For example, if you specify the path `/input` and enter `10.10.10.10` for the URL, requests for `10.10.10.10/input` will be sent from the destination.
Buffer flush interval	Time (in seconds) between sending batches of data to the destination. The default value is `1 second`.
Number of handlers	The number of services that are processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
Health check path	The URL for sending requests to obtain health information about the system that the destination resource is connecting to.
Health check timeout	Frequency of the health check in seconds.
Health Check Disabled	Check box that disables the health check.
Debug	This toggle switch lets you specify whether resource logging must be enabled. The default value is **Disabled**.
Disk buffer disabled	Drop-down list that lets you enable or disable the disk buffer. By default, the disk buffer is enabled. The disk buffer is used if the collector cannot send normalized events to the destination. The amount of allocated disk space is limited by the value of the **Disk buffer size limit** setting. If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer.
Filter	In the **Filter** section, you can specify the criteria for identifying events that must be processed by the resource. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources**

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. The name must contain 1 to 128 Unicode characters.
4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.

- **<**—the left operand is less than the right operand.

- **<=**—the left operand is less than or equal to the right operand.

- **>**—the left operand is greater than the right operand.

- **>=**—the left operand is greater than or equal to the right operand.

- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).

- **contains**—the left operand contains values of the right operand.

- **startsWith**—the left operand starts with one of the values of the right operand.

- **endsWith**—the left operand ends with one of the values of the right operand.

- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.

- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.

- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.

- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.

- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.

- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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.

Diode type

The **diode** type is used to transmit events using a data diode.

Basic settings tab

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
The **State** toggle switch	Used when events must be sent to the destination. By default, sending events is enabled.
Type	Required setting. Destination type, **diode**.
Data diode source directory	Required setting. The directory from which the data diode moves events. The path can contain up to 255 Unicode characters.

Limitations when using prefixes in paths on Windows servers

On Windows servers, absolute paths to directories must be specified. Directories with names matching the following regular expressions cannot be used:

- `^[a-zA-Z]:\\Program Files`
- `^[a-zA-Z]:\\Program Files \(x86\)`
- `^[a-zA-Z]:\\Windows`
- `^[a-zA-Z]:\\Program Files\\Kaspersky Lab\\KUMA`

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/

Files are available at the following paths:

- /opt/kaspersky/kuma/clickhouse/logs/

	• /opt/kaspersky/kuma/mongodb/log/ • /opt/kaspersky/kuma/victoria-metrics/log/
Temporary directory	Directory in which events are prepared for transmission to the data diode. Events are stored in a file when a timeout (10 seconds by default) or a buffer overflow occurs. The prepared file is moved to the directory specified in the **Data diode source directory** field. The checksum (SHA-256) of the file contents is used as the name of the file containing events. The temporary directory must be different from the data diode source directory.
Description	Resource description: up to 4,000 Unicode characters.

Advanced settings tab

Setting	Description
Compression	You can use Snappy compression. By default, compression is **disabled**. This setting must match for the connector and destination resources used to relay events from an isolated network segment via the data diode.
Buffer size	Sets the size of the buffer. The default value is 1 KB, and the maximum value is 64 MB.
Delimiter	In the drop-down list, you can select the character to mark the boundary between events. By default, `\n` is used. This setting must match for the connector and destination resources used to relay events from an isolated network segment via the data diode.
Buffer flush interval	Time (in seconds) between sending batches of data to the destination. The default value is `1 second`.
Number of handlers	This field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
Debug	This toggle switch lets you specify whether resource logging must be enabled. The default value is **Disabled**.
Filter	In the **Filter** section, you can specify the criteria for identifying events that must be processed by the resource. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources**

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. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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.

Kafka type

The **kafka** type is used for Kafka communications.

Basic settings tab

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
The **State** toggle switch	Used when events must be sent to the destination. By default, sending events is enabled.
Type	Required setting. Destination type, **kafka**.
URL	Required setting.

	URL that you want to connect to. Available formats: `host:port`, `IPv4:port`, `:port`. IPv6 addresses are also supported, however, when you use them, you must specify the interface as well: `[address%interface]:port`. Example: `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`). You can add multiple addresses using the URL button.
Topic	Required setting. Subject of Kafka messages. Must contain from 1 to 255 of the following characters: a–z, A–Z, 0–9, ".", "_", "-".
Delimiter	Specify a character that defines where one event ends and the other begins. By default, `\n` is used.
Authorization	Type of authorization when connecting to the specified URL Possible values: - **disabled** is the default value. - **PFX —** a certificate must be generated with a private key in PKCS#12 container format in an external Certificate Authority. Then the certificate must be exported from the key store and uploaded to the KUMA web interface as a PFX secret. - Add PFX secret 1. If you previously uploaded a PFX certificate, select it from the **Secret** drop-down list. If no certificate was previously added, the drop-down list shows **No data**. 2. If you want to add a new certificate, click the + button on the right of the **Secret** list. The **Secret** window opens. 3. In the **Name** field, enter the name that will be used to display the secret in the list of available secrets. 4. Click the **Upload PFX** button to select the file containing your previously exported certificate with a private key in PKCS#12 container format. 5. In the **Password** field, enter the certificate security password that was set in the Certificate Export Wizard. 6. Click the **Save** button. The certificate will be added and displayed in the **Secret** list. - **plain** — you must indicate the secret containing user account credentials for authorization when connecting to the connector. Add secret

	1. If you previously created a secret, select it from the **Secret** drop-down list. If no secret was previously added, the drop-down list shows **No data**. 2. If you want to add a new secret, click the + button on the right of the **Secret** list. The **Secret** window opens. 3. In the **Name** field, enter the name that will be used to display the secret in the list of available secrets. 4. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector. 5. If necessary, add any other information about the secret in the **Description** field. 6. Click the **Save** button. The secret will be added and displayed in the **Secret** list.
Description	Resource description: up to 4,000 Unicode characters.

Advanced settings tab

Setting	Description
Buffer size	Sets the size of the buffer. The default value is 1 KB, and the maximum value is 64 MB.
Timeout	The time (in seconds) to wait for a response from another service or component. The default value is `30`.
Disk buffer size limit	Size of the disk buffer in bytes. The default value is 10 GB.
Output format	Format for sending events to an external destination. Available values: • JSON • CEF
TLS mode	Use of TLS encryption. Available values: • **Disabled** (default) means TLS encryption is not used. • **Enabled** means encryption is used, but the certificate is not verified. • **With verification** means encryption is used with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during program installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/.

- **Custom CA** means encryption is used with verification that the certificate was signed by a Certificate Authority. The secret containing the certificate is selected from the **Custom CA** drop-down list, which is displayed when this option is selected.
  Creating a certificate signed by a Certificate Authority

  > To use this TLS mode, you must do the following on the KUMA Core server (OpenSSL commands are used in the examples below):
  >
  > 1. Create the key that will be used by the Certificate Authority.
  >
  >   Example command:
  >
  >    ```
  >    openssl genrsa -out ca.key 2048
  >    ```
  >
  > 2. Generate a certificate for the key that was just created.
  >
  >   Example command:
  >
  >    ```
  >    openssl req -new -x509 -days 365 -key ca.key -subj "/CN=<common
  >    host name of Certificate Authority>" -out ca.crt
  >    ```
  >
  > 3. Create a private key and a request to have it signed by the Certificate Authority.
  >
  >   Example command:
  >
  >    ```
  >    openssl req -newkey rsa:2048 -nodes -keyout server.key -subj "/CN=
  >    <common host name of KUMA server>" -out server.csr
  >    ```
  >
  > 4. Create a certificate signed by the Certificate Authority. The subjectAltName must include the domain names or IP addresses of the server for which the certificate is being created.
  >
  >   Example command:
  >
  >    ```
  >    openssl x509 -req -extfile <(printf
  >    "subjectAltName=DNS:domain1.ru,DNS:domain2.com,IP:192.168.0.1") -
  >    days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -
  >    out server.crt
  >    ```
  >
  > 5. The obtained `server.crt` certificate should be uploaded in the KUMA web interface as a **certificate**-type secret, which should then be selected from the **Custom CA** drop-down list.

  When using TLS, it is impossible to specify an IP address as a URL.

Delimiter	In the drop-down list, you can select the character to mark the boundary between events. By default, `\n` is used.
Buffer flush interval	Time (in seconds) between sending batches of data to the destination. The default value is `1 second`.
Number of handlers	This field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
Debug	This toggle switch lets you specify whether resource logging must be enabled. The default value is **Disabled**.
Disk buffer disabled	Drop-down list that lets you enable or disable the disk buffer. By default, the disk buffer is enabled. The disk buffer is used if the collector cannot send normalized events to the destination. The amount of allocated disk space is limited by the value of the **Disk buffer size limit** setting.

	If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer.
Filter	In this section, you can specify the criteria for identifying events that must be processed by the resource. You can select an existing filter from the drop-down list or **create** a new filter. Creating a filter in resources ?

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. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In

other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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.

File type

The **file** type is used for writing data to a file.

If you delete a destination of the 'file' type used in a service, that service must be restarted.

Basic settings tab

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
The **State** toggle switch	Used when events must be sent to the destination. By default, sending events is enabled.
Type	Required setting. Destination type, **file**.
URL	Required setting. Path to the file to which the events must be written. Limitations when using prefixes in file paths

Prefixes that cannot be used when specifying paths to files:

- /*
- /bin
- /boot
- /dev
- /etc
- /home
- /lib
- /lib64
- /proc
- /root
- /run
- /sys
- /tmp
- /usr/*
- /usr/bin/
- /usr/local/*
- /usr/local/sbin/
- /usr/local/bin/
- /usr/sbin/
- /usr/lib/
- /usr/lib64/
- /var/*
- /var/lib/
- /var/run/
- /opt/kaspersky/kuma/

Files are available at the following paths:

- /opt/kaspersky/kuma/clickhouse/logs/

	- /opt/kaspersky/kuma/mongodb/log/ - /opt/kaspersky/kuma/victoria-metrics/log/
Description	Resource description: up to 4,000 Unicode characters.

Advanced settings tab

Setting	Description
Buffer size	Sets the size of the buffer. The default value is 1 KB, and the maximum value is 64 MB.
Disk buffer size limit	Size of the disk buffer in bytes. The default value is 10 GB.
Delimiter	In the drop-down list, you can select the character to mark the boundary between events. \n is used by default.
Buffer flush interval	Time (in seconds) between sending batches of data to the destination. The default value is `1 second`.
Number of handlers	This field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
Output format	Format for sending events to an external destination. Available values: - JSON - CEF
Debug	This toggle switch lets you specify whether resource logging must be enabled. The default value is **Disabled**.
Disk buffer disabled	Drop-down list that lets you enable or disable the disk buffer. By default, the disk buffer is enabled. The disk buffer is used if the collector cannot send normalized events to the destination. The amount of allocated disk space is limited by the value of the **Disk buffer size limit** setting. If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer.
Filter	In the **Filter** section, you can specify the criteria for identifying events that must be processed by the resource. You can select an existing filter from the drop-down list or **create** a new filter. Creating a filter in resources

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. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In

other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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.

Storage type

The **storage** type is used to transmit data to the storage.

Basic settings tab

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
The **State** toggle switch	Used when events must be sent to the destination. By default, sending events is enabled.
Type	Required setting. Destination type, **storage**.
URL	Required setting. URL that you want to connect to. Available formats: `host:port`, `IPv4:port`, `:port`. IPv6 addresses are also supported, however, when you use them, you must specify the interface as well: `[address%interface]:port`. Example: `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`). You can add multiple addresses using the **URL** button.

	The **URL** field supports search for services by FQDN, IP address, and name. Search string formats: • `<Search value>`—search is performed by FQDN, IP addresses, and service names. • `<First search value ending in one or more digits>:<second search value>`—the first value is used to search by the service FQDN or IP address, and the second value is used to search by port. • `:<value>`—search is performed by port.
Description	Resource description: up to 4,000 Unicode characters.

Advanced settings tab

Setting	Description
Proxy server	Drop-down list for selecting a proxy server.
Buffer size	Sets the size of the buffer. The default value is 1 KB, and the maximum value is 64 MB.
Disk buffer size limit	Size of the disk buffer in bytes. The default value is 10 GB.
URL selection policy	Drop-down list in which you can select a method for determining which URL to send events to if several URLs have been specified: • **Any**. Events are sent to one of the available URLs as long as this URL receives events. If the connection is broken (for example, the receiving node is disconnected) a different URL will be selected as the events destination. • **Prefer first**. Events are sent to the first URL in the list of added addresses. If it becomes unavailable, events are sent to the next available node in sequence. When the first URL becomes available again, events start to be sent to it again. • **Balanced** means that packages with events are evenly distributed among the available URLs from the list. Because packets are sent either on a destination buffer overflow or on the flush timer, this URL selection policy does not guarantee an equal distribution of events to destinations.
Buffer flush interval	Time (in seconds) between sending batches of data to the destination. The default value is **1** `second`.
Number of handlers	This field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
Health check timeout	Frequency of the health check in seconds.
Debug	This toggle switch lets you specify whether resource logging must be enabled. The default value is **Disabled**.
Disk buffer disabled	Drop-down list that lets you enable or disable the disk buffer. By default, the disk buffer is enabled.

	The disk buffer is used if the collector cannot send normalized events to the destination. The amount of allocated disk space is limited by the value of the **Disk buffer size limit** setting. If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer.
Filter	In this section, you can specify the criteria for identifying events that must be processed by the resource. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources** ?

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. The name must contain 1 to 128 Unicode characters.
4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

   Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

   Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In

other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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.

Correlator type

The **correlator** type is used to transmit data to the correlator.

Basic settings tab

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
The **State** toggle switch	Used when events must be sent to the destination. By default, sending events is enabled.
Type	Required setting. Destination type, **correlator**.
URL	Required setting. URL that you want to connect to. Available formats: `host:port`, `IPv4:port`, `:port`. IPv6 addresses are also supported, however, when you use them, you must specify the interface as well: `[address%interface]:port`. Example: `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`). You can add multiple addresses using the **URL** button.

	The **URL** field supports search for services by FQDN, IP address, and name. Search string formats: • `<Search value>`—search is performed by FQDN, IP addresses, and service names. • `<First search value ending in one or more digits>:<second search value>`—the first value is used to search by the service FQDN or IP address, and the second value is used to search by port. • `:<value>`—search is performed by port.
Description	Resource description: up to 4,000 Unicode characters.

Advanced settings tab

Setting	Description
Proxy server	Drop-down list for selecting a proxy server.
Buffer size	Sets the size of the buffer. The default value is 1 KB, and the maximum value is 64 MB.
Disk buffer size limit	Size of the disk buffer in bytes. The default value is 10 GB.
URL selection policy	Drop-down list in which you can select a method for determining which URL to send events to if several URLs have been specified: • **Any**. Events are sent to one of the available URLs as long as this URL receives events. If the connection is broken (for example, the receiving node is disconnected) a different URL will be selected as the events destination. • **Prefer first**. Events are sent to the first URL in the list of added addresses. If it becomes unavailable, events are sent to the next available node in sequence. When the first URL becomes available again, events start to be sent to it again. • **Balanced** means that packages with events are evenly distributed among the available URLs from the list. Because packets are sent either on a destination buffer overflow or on the flush timer, this URL selection policy does not guarantee an equal distribution of events to destinations.
Buffer flush interval	Time (in seconds) between sending batches of data to the destination. The default value is `1 second`.
Number of handlers	This field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
Health check timeout	Frequency of the health check in seconds.
Debug	This toggle switch lets you specify whether resource logging must be enabled. The default value is **Disabled**.
Disk buffer disabled	Drop-down list that lets you enable or disable the disk buffer. By default, the disk buffer is enabled.

	The disk buffer is used if the collector cannot send normalized events to the destination. The amount of allocated disk space is limited by the value of the **Disk buffer size limit** setting. If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer.
Filter	In the **Filter** section, you can specify the criteria for identifying events that must be processed by the resource. You can select an existing filter from the drop-down list or **create** a new filter. Creating a filter in resources ?

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. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.

- **<**—the left operand is less than the right operand.

- **<=**—the left operand is less than or equal to the right operand.

- **>**—the left operand is greater than the right operand.

- **>=**—the left operand is greater than or equal to the right operand.

- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).

- **contains**—the left operand contains values of the right operand.

- **startsWith**—the left operand starts with one of the values of the right operand.

- **endsWith**—the left operand ends with one of the values of the right operand.

- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.

- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.

- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.

- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.

- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.

- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In

other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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.

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.

Working with events

In the **Events** section of the KUMA web interface, you can inspect events received by the program 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 application 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 🔍 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 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 **Events** section of the KUMA web interface, click the relevant event.

     The **Event details** area appears in the right part of the window.

  2. Change the filter settings by using the plus or minus icons next to the relevant settings:

     - 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.

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** 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. See also 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 and select the relevant storage cluster.

Now events from the selected storage are displayed in the events table. The name of the selected storage is displayed in the ☷ drop-down list.

The ☷ drop-down list displays only the clusters of tenants available to the user, and the cluster of the main tenant.

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 are using aggregation functions in a query, you cannot customize the events table display, sort events in ascending or descending order, or receive statistics.

   > 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 a group of conditions, but the operator can be changed by clicking on this value. 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.

     Filter conditions can be deleted by using the ✕ button. Group conditions are deleted using the **Delete group** button.

   - **GROUP BY**—event fields or aliases to be used for grouping the returned data.

If you are using data grouping in a query, you cannot customize the events table display, sort events in ascending or descending order, receive statistics, or perform a retroscan.

> 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 records** button to display additional strings in the table. This button is not displayed when filtering events by the standard period.

3. Click **Apply**.

   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.

4. Click the 🔍 button to display the data in the table.

The table will display the search results based on the generated SQL query.

> 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 string to manually create SQL queries of any complexity for filtering 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 queries.

3. Click the 🔍 button.

You will see a table of events that satisfy the criteria of your query. If necessary, you can filter events by period.

Supported functions and operators

- `SELECT`—event fields that should be returned.

  For `SELECT` fields, the program supports the following functions and operators:

  - Aggregation functions: `count, avg, max, min, sum`.
  - Arithmetic operators: `+, -, *, /, <, >, =, !=, >=, <=`.

    You can combine these functions and operators.

    If you are using aggregation functions in a query, you cannot customize the events table display, sort events in ascending or descending order, or receive statistics.

- `DISTINCT`—removes duplicates from the result of a SELECT statement. You must use the following notation: SELECT DISTINCT SourceAddress as Addresses FROM `<rest of the query>`.

- `FROM`—data source.

  > When creating a 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 queries, special characters must be shielded with "\")

- `GROUP BY`—event fields or aliases to be used for grouping the returned data.

  If you are using data grouping in a query, you cannot customize the events table display, sort events in ascending or descending order, receive statistics, or perform a retroscan.

- `ORDER BY`—columns used as the basis for sorting the returned data.

  Possible values:

  - `DESC`—descending order.
  - `ASC`—ascending order.

- `OFFSET`—skip the indicated number of lines before printing the query results output.

- `LIMIT`—number of strings displayed in the table.

  The default value is 250.

  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 records** button to display additional strings in the table. This button is not displayed when filtering events by the standard period.

Example queries:

- `SELECT * FROM `events` WHERE Type IN ('Base', 'Audit') ORDER BY Timestamp DESC LIMIT 250`
  In the events table, all events with the **Base** and **Audit** type are sorted by the **Timestamp** column in descending order. The number of strings that can be displayed in the table is 250.

- `SELECT * FROM `events` WHERE BytesIn BETWEEN 1000 AND 2000 ORDER BY Timestamp ASC LIMIT 250`
  All events of the events table for which the **BytesIn** field contains a value of received traffic in the range from 1,000 to 2,000 bytes are sorted by the **Timestamp** column in ascending order. The number of strings that can be displayed in the table is 250.

- `SELECT * FROM `events` WHERE Message LIKE '%ssh:%' ORDER BY Timestamp DESC LIMIT 250`
  In the events table, all events whose **Message** field contains data corresponding to the defined `%ssh:%` template in lowercase are sorted by the **Timestamp** column in descending order. The number of strings that can be displayed in the table is 250.

- `SELECT * FROM `events` WHERE inSubnet(DeviceAddress, '00.0.0.0/00') ORDER BY Timestamp DESC LIMIT 250`
  In the events table, all events for the hosts that are in the 00.0.0.0/00 subnet are sorted by the **Timestamp** column in descending order. The number of strings that can be displayed in the table is 250.

- `SELECT * FROM `events` WHERE match(Message, 'ssh.*') ORDER BY Timestamp DESC LIMIT 250`
  In the events table, all events whose **Message** field contains text corresponding to the `ssh.*` template are sorted by the **Timestamp** column in descending order. The number of strings that can be displayed in the table is 250.

- `SELECT max(BytesOut) / 1024 FROM `events``
  Maximum amount of outbound traffic (KB) for the selected time period.

- `SELECT count(ID) AS "Count", SourcePort AS "Port" FROM `events` GROUP BY SourcePort ORDER BY Port ASC LIMIT 250`
  Number of events and port number. Events are grouped by port number and sorted by the **Port** column in ascending order. The number of strings that can be displayed in the table is 250.
  The **ID** column in the events table is named Count, and the **SourcePort** column is named Port.

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.

When switching to the query builder, the query parameters that were manually entered into the search string are not transferred to the builder so you will need to create your query again. Also, the query created in the builder does not overwrite the query that was entered into the search string until you click the **Apply** button in the builder window.

Aliases must not contain spaces.

For more details on SQL, refer to the ClickHouse documentation. See also the supported ClickHouse functions.

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, open the **Period** drop-down list in the upper part of the window.

2. If you want to filter events based on a standard period, select one of the following:

   - **5 minutes**

   - **15 minutes**

   - **1 hour**

   - **24 hours**

   - **In period**

     If you select this option, use the opened calendar to select the start and end dates of the period and click **Apply Filter**. The date and time format depends on your operating system's settings. You can also manually change the date values if necessary.

3. Click the 🔍 button.

   When the period filter is set, only events registered during the specified time interval will be 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 series or select the relevant time period and click the **Show events** button.

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`
`SeriviceID`	`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 the window that opens, 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 🔍 to execute 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 the **Related to alert** option is selected in the ⊞ drop-down list. If this is the case, only the functions and operators listed below are available for event filtering.

> If the **All events** option is selected from the ⊞ drop-down list, these limitations are not applied.

- `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 records** 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 and select **Save current filter**.
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. In the **Tenant** drop-down list, select the tenant that will own the created filter.
4. 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 🔍 button to submit the search query.

You can click the ☆ icon near the filter configuration name to make it a default filter.

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 and click the 🗑 icon next to the configuration that you need to delete.
2. Click **OK**.

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 program 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.
- Clicking a file hash in the **FileHash** field opens 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 program.
- You can 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 program is in alert investigation mode and open the **Correlation event details** window if the selected event is a correlation event.

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
- SeriviceID
- 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, open the ⋯ drop-down list and choose **Export TSV**.

   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 download, the ✓ icon will appear in the **Status** column of the task.

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:

- **Tenant**.
- **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.

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.

   You will see a window for selecting the event fields that should be displayed in the events table.

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.

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 the update frequency in the ↻ drop-down list:

- **5 seconds**
- **15 seconds**
- **30 seconds**
- **1 minute**
- **5 minutes**
- **15 minutes**

The events table now refreshes automatically.

To disable automatic refresh:

Select **No refresh** in the ↻ drop-down list:

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. Relevant parameters can be found by using the **Search** 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. If the selected event is a correlation event, the **Detailed view** button will be displayed at the bottom of the details area.

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.

The **Correlation event details** section of the correlation event window contains the following data:

- **Correlation event severity**—the importance 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 importance 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.

Normalizers

Normalizers are used for converting raw events that come from various sources in different formats to the KUMA event data model. Normalized events become available for processing by other KUMA resources and services.

A normalizer consists of the *main* event parsing rule and optional *additional event parsing rules*. By creating a main parsing rule and a set of additional parsing rules, you can implement complex event processing logic. Data is passed along the tree of parsing rules depending on the conditions specified in the

Extra normalization conditions setting. The sequence in which parsing rules are created is significant: the event is processed sequentially and the processing sequence is indicated by arrows.
The following event normalization options are now available:

- 1 collector — 1 normalizer

  We recommend using this method if you have many events of the same type or many IP addresses from which events of the same type may originate. You can configure one collector with only one normalizer, which is optimal in terms of performance.

- 1 collector — multiple normalizers linked to IP

  This method is available for collectors with a connector of UDP, TCP, or HTTP type. If a UDP, TCP, or HTTP connector is specified in the collector at the 'Transport' step, then at the 'Event parsing' step, you can specify multiple IP addresses on the 'Parsing settings' tab and choose the normalizer that you want to use for events coming from the specified addresses. The following types of normalizers are available: json, cef, regexp, syslog, csv, kv, xml. For normalizers of the Syslog and regexp types, you can specify extra normalization conditions depending on the value of the DeviceProcessName field.

A normalizer is created in several steps:

1. **Preparing to create a normalizer**

   A normalizer can be created in the 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 the mouse over the circle or the block corresponding to the normalizer opens the **Additional event parsing** window where you can specify the settings of the additional parsing rule:

- Specify the conditions for sending data to the new normalizer.
- Specify event parsing settings.
- Specify event enrichment settings.

The additional event parsing rule is displayed in the normalizer as a dark block. The block displays the triggering conditions for the additional parsing rule, the name of the additional parsing rule, and the event field. When this event field is available, the data is passed to the normalizer. Click the block of the additional parsing rule to view or modify its settings.

If you hover the mouse over the additional normalizer, a plus button appears. You can use this button to create a new additional event parsing rule. To delete a normalizer, use the button with the trash icon.

4 **Completing the creation of the normalizer**

To finish the creation of the normalizer, click **Save**.

In the upper right corner, in the search field, you can search for additional parsing rules by name.

For normalizer resources, you can enable the display of control characters in all input fields except the **Description** field.

If, when changing the settings of a collector resource set, you change or delete conversions in a normalizer connected to it, the edits will not be saved, and the normalizer itself may be corrupted. If you need to modify conversions in a normalizer that is already part of a service, the changes must be made directly to the normalizer under **Resources** → **Normalizers** in the web interface.

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 settings:

- **Name** (required)—name of the parsing rules. Must contain 1 to 128 Unicode characters. The name of the main parsing rule is used as the name of the normalizer.
- **Tenant** (required)—name of the tenant that owns the resource.

This setting is not available for extra parsing rules.

- **Parsing method** (required)—drop-down list for selecting the type of incoming events. Depending on your choice, you can use the preconfigured rules for matching event fields or set your own rules. When you select some parsing methods, additional parameter fields required for filling in may become available.

  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 arranged data, you can access the fields of nested objects by specifying the names of the parameters dividing them by a period. 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 additional normalizer in its entirety.

    Newline characters can be `\n` and `\r\n`. Strings must be UTF-8 encoded.

    If you want to send the raw event for advanced normalization, at each nesting level in the **Advanced event parsing** window, select **Yes** in the **Keep raw event** drop-down list.

  - cef

    This parsing method is used to process CEF data.

    When choosing this method, you can use the preconfigured rules for converting events to the KUMA format by clicking the **Apply default mapping** button.

  - regexp

This parsing method is used to create custom rules for processing data in a format using regular expressions.

In the **Normalization** parameter block field, add a regular expression (RE2 syntax) with named capture groups. The name of a group and its value will be interpreted as the field and the value of the raw event, which can be converted into an event field in KUMA format.

To add event handling rules:

1. Copy an example of the data you want to process to the **Event examples** field. This is an optional but recommended step.
2. In the **Normalization** parameter block field add a regular expression with named capture groups in RE2 syntax, for example "(?P<name>regexp)". The regular expression added to the **Normalization** parameter must exactly match the event. Also, when developing the regular expression, it is recommended to use special characters that match the starting and ending positions of the text: ^, $.

   You can add multiple regular expressions by using the **Add regular expression** button. If you need to remove the regular expression, use the ✕ button.
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. Now you can select the corresponding KUMA field in the column next to each capture group. Otherwise, if you named the capture groups in accordance with the CEF format, you can use the automatic CEF mapping by selecting the **Use CEF syntax for normalization** check box.

Event handling rules were added.

- syslog ?

  This parsing method is used to process data in syslog format.

  When choosing this method, you can use the preconfigured rules for converting events to the KUMA format by clicking the **Apply default mapping** button.

- csv ?

  This parsing method is used to create custom rules for processing CSV data.

  When choosing this 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.

- kv ?

This parsing method is used to process data in key-value pair format.

If you select this method, you must provide values in the following required fields:

- **Pair delimiter**—specify a character that will serve as a delimiter for key-value pairs. You can specify any one-character (1 byte) value, provided that the character does not match the value delimiter.
- **Value delimiter**—specify a character that will serve as a delimiter between the key and the value. You can specify any one-character (1 byte) value, provided that the character does not match the delimiter of key-value pairs.

- **xml** ?

This parsing method is used to process XML data in which each object, including its nested objects, occupies a single line in a file. Files are processed line by line.

If you want to send the raw event for advanced normalization, at each nesting level in the **Advanced event parsing** window, select **Yes** in the **Keep raw event** drop-down list.

When this method is selected in the parameter block **XML attributes** you can specify the key attributes to be extracted from tags. If an XML structure has several attributes with different values in the same tag, you can indicate the necessary value by specifying its key in the **Source** column of the **Mapping** table.

To add key XML attributes,

Click the **Add field** button, and in the window that appears, specify the path to the required attribute.

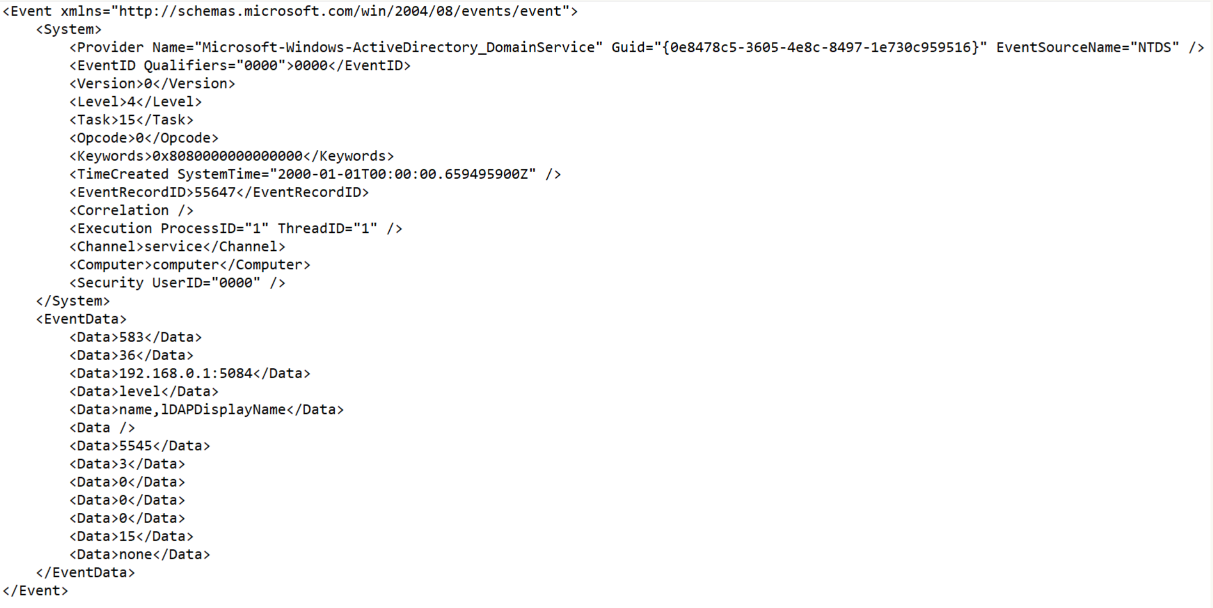
You can add more than one attribute. Attributes can be removed one at a time using the cross icon or all at once using the **Reset** button.

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

Tag numbering is available as of KUMA 2.1.3. This functionality allows automatically numbering tags in XML events, which lets you parse an event with identical tags or unnamed tags, such as <Data>.

As an example, we will use the **Tag numbering** functionality to number the tags of the EventData attribute of 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>0</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 such events, you must:

- Configure tag numbering.
- Configure data mapping for numbered tags with KUMA event fields.

Simultaneous use of **XML attributes** and **Tag numbering** leads to incorrect operation of the normalizer. If an attribute contains unnamed tags or the identical tags, we recommend using the **Tag numbering** functionality. If the attribute contains only named tags, use **XML attributes**.

To configure parsing of events with identically named or unnamed tags:

1. Create a new normalizer or open an existing normalizer for editing.
2. In the **Basic event parsing** window of the normalizer, in the **Parsing method** drop-down list, select 'xml' and in the **Tag numbering** field, click **Add field**.

   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 number to be assigned to a tag is 0. If the tag is empty, for example, <Data />, it is also assigned a number.
3. To configure data mapping, under **Mapping**, click **Add row** and do the following:
   a. In the new row, in the **Source** field, enter the full path to the tag and its index. For the Microsoft Windows event from the example above, the full path with indices look like this:
      - 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.
4. To save changes:
   - If you created a new normalizer, click **Save**.
   - If you edited an existing normalizer, click **Update configuration** in the collector to which the normalizer is linked.

Parsing is configured.

- netflow5

This parsing method is used to process data in the NetFlow v5 format.

When choosing this method, you can use the preconfigured rules for converting events to the KUMA format by clicking the **Apply default mapping** button. If the netflow5 type is selected for the main parsing, extra normalization is not available.

> In mapping rules, the protocol type for **netflow5** is not indicated in the fields of KUMA events by default. 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.
  >
  > When choosing this method, you can use the preconfigured rules for converting events to the KUMA format by clicking the **Apply default mapping** button. If the netflow9 type is selected for the main parsing, extra normalization is not available.
  >
  > > In mapping rules, the protocol type for **netflow9** is not indicated in the fields of KUMA events by default. 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.
    >
    > When choosing this method, you can use the preconfigured rules for converting events to the KUMA format by clicking the **Apply default mapping** button. If the sflow5 type is selected for the main parsing, extra normalization is not available.

  - ipfix

    > This parsing method is used to process IPFIX data.
    >
    > When choosing this method, you can use the preconfigured rules for converting events to the KUMA format by clicking the **Apply default mapping** button. If the ipfix type is selected for the main parsing, extra normalization is not available.
    >
    > > In mapping rules, the protocol type for **ipfix** is not indicated in the fields of KUMA events by default. 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 method to process data obtained by making a selection from the database.

- **Keep raw event** (required)—in this drop-down list, indicate whether you need to store the raw event in the newly created normalized event. Available values:

  - **Don't save**—do not save the raw event. This is the default setting.

  - **Only errors**—save the raw event in the `Raw` field of the normalized event if errors occurred when parsing it. This value is convenient to use when debugging a service. In this case, every time an event has a non-empty `Raw` field, you know there was a problem.

    > If fields containing the names `*Address` or `*Date*` do not comply with normalization rules, these fields are ignored. No normalization error occurs in this case, and the values of the fields are not displayed in the `Raw` field of the normalized event even if the **Keep raw event** → **Only errors** option was selected.

  - **Always**—always save the raw event in the `Raw` field of the normalized event.

This setting is not available for extra parsing rules.

- **Keep extra fields** (required)—in this drop-down list, you can choose whether you want to save fields and their values if no mapping rules have been configured for them (see below). 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`.

      A value of this type is specified 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.

      A value of this type is specified manually.

      To work with a value from the Extra field at depth 3 and below, use backquotes ``. 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.

- **Description**—resource description: up to 4,000 Unicode characters.

  This setting is not available for extra parsing rules.

- **Event examples**—in this field, you can provide an example of data that you want to process.

  This setting is not available for the following parsing methods: **netflow5**, **netflow9**, **sflow5**, **ipfix**, **sql**.

  The **Event examples** field is populated with data obtained from the raw event if the event was successfully parsed and the type of data obtained from the raw event matches the type of the KUMA field.

  For example, the value "192.168.0.1" enclosed in quotation marks is not displayed in the SourceAddress field, in this case the value 192.168.0.1 is displayed in the **Event examples** field.

- **Mapping** settings block—here you can configure mapping of raw event fields to fields of the event in KUMA format:

- **Source**—column for the names of the raw event fields that you want to convert into KUMA event fields.

  Clicking the 🔧 button next to the field names in the **Source** column opens the **Conversion** window, in which you can use the **Add conversion** button to create rules for modifying the original data before they are written to the KUMA event fields. 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 changes that can be applied to a value before it gets written to the event field. The conversion type is selected from a drop-down list.

Available conversions:

- **lower**—is used to make all characters of the value lowercase
- **upper**—is used to make all characters of the value uppercase
- **regexp** – used to convert a value using the regular expression RE2. When this conversion type is selected, the field appears where regular expression should be added.
- **substring**—is used to extract characters in the position range specified in the **Start** and **End** fields. These fields appear when this conversion type is selected.
- **replace**—is used to replace specified character sequence with the other character sequence. When this type of conversion is selected, new fields appear:
  - **Replace chars**—in this field you can specify the character sequence that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- **trim**—used to simultaneously remove the characters specified in the **Chars** field from the leading and end positions of the value. The field appears when this type of conversion is selected. For example, a **trim** conversion with the `Micromon` value applied to `Microsoft-Windows-Sysmon` results in `soft-Windows-Sys`.
- **append** is used to add the characters specified in the **Constant** field to the end of the event field value. The field appears when this type of conversion is selected.
- **prepend**—used to prepend the characters specified in the **Constant** field to the start of the event field value. The field appears when this type of conversion is selected.
- **replace with regexp**—is used to replace RE2 regular expression results with the character sequence.
  - **Expression**—in this field you can specify the regular expression which results that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, such a string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the "String" type, all types of conversions are available.
- For fields of the "Number" and "Float" types, the following types of conversions are available: regexp, substring, replace, trim, append, prepend, replaceWithRegexp, decodeHexString, decodeBase64String, decodeBase64URLString.
- For fields of "Array of strings", "Array of numbers", and "Array of floats" types, the following types of conversions are available: append, prepend.

- **KUMA field**—drop-down list for selecting the required fields of KUMA events. You can search for fields by entering their names in the field.
- **Label**—in this column, you can add a unique custom label to event fields that begin with `DeviceCustom*` and `Flex*`.

New table rows can be added by using the **Add row** button. Rows can be deleted individually using the ✕ button or all at once using the **Clear all** button.

If you have loaded data into the **Event examples** field, the table will have an **Examples** column containing examples of values carried over from the raw event field to the KUMA event field.

If the size of the KUMA event field is less than the length of the value placed in it, the value is truncated to the size of the event field.

Extended event schema

When normalizing events, extended event schema fields can be used in addition to standard KUMA event schema fields. Information about the types of extended event schema fields is shown in the table below.

Using many unique fields of the extended event schema can reduce the performance of the system, increase the amount of disk space required for storing events, and make the information difficult to understand.

We recommend consciously choosing a minimal set of additional fields of the extended event schema that you want to use in normalizers and correlation.

To use extended event schema fields:

- Open an existing event normalizer or create a new event normalizer.
- Specify the basic settings of the normalizer.
- Click "Add row".
- For the "Source" setting, enter the name of the source field in the raw event.
- For the "KUMA field" setting, enter the name of the extended event schema field that you are creating (see the table below). You can also use an existing field of the extended event schema.

Fields of the extended data model of normalized events:

Field name Specified in the KUMA field setting	Data type	Availability in the normalizer	Description
S.`<field name>`	String	All types	Field of the "String" type
N.`<field name>`	Number	All types	Field of the "Number" type
F.`<field name>`	Float	All types	Field of the "Float" type
SA.`<field name>`	Array of strings	KV, JSON	Field of the "Array of strings" type The order of the array elements is the same as the order of the elements of the raw event.
NA.`<field name>`	Array of integers	KV, JSON	A field of the "Array of integers" type. The order of the array elements is the same as the order of the elements of the raw event.
FA.`<field name>`	Array of floats	KV, JSON	Field of the "Array of floats" type The order of the array elements is the same as the order of the elements of the raw event.

The prefixes "S.", "N.", "F.", "SA.", "NA.", "FA." are required when creating fields of the extended event schema; the prefixes must be strictly uppercase.

Replace <field name> with the field name. You may use letters of the English alphabet and numerals in the field name. The space character is not allowed.

- Click OK.

- Click Save to finish editing the event normalizer.

The normalizer is saved, and the additional field is created. After saving the normalizer, the additional field can be used in other normalizers.

Note: If the data in the fields of the raw event does not match the type of the KUMA field, the value is not saved during the normalization of events. For example, the string "test" cannot be written to the DeviceCustomNumber1 KUMA field of the Number type.

If you want to minimize the load on the storage server when searching events, preparing reports, and performing other operations on events in storage, use KUMA event schema fields as your first preference, extended event schema fields as your second preference, and the Extra fields as your last resort.

Enrichment in the normalizer

When creating event parsing rules in the normalizer settings window, on the **Enrichment** tab, you can configure the rules for adding extra data to the fields of the normalized event using enrichment rules. These enrichment rules are stored in the settings of the normalizer where they were created.

Enrichments are created by using the **Add enrichment** button. There can be more than one enrichment rule. You can delete enrichment rules by using the × button. Extended event schema fields can be used for event enrichment.

Settings available in the enrichment rule settings block:

- **Source kind** (required)—drop-down list for selecting the type of enrichment. Depending on the selected type, you may see advanced settings that will also need to be completed.

  Available Enrichment rule source types:

  - constant ?

    This type of enrichment is used when a constant needs to be added to an event field. Settings of this type of enrichment:

    - In the **Constant** field, specify the value that should be added to the event field. The value may not be longer than 255 Unicode characters. If you leave this field blank, the existing event field value will be cleared.
    - In the **Target field** drop-down list, select the KUMA event field to which you want to write 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.

    When this type is selected in the **Dictionary name** drop-down list, you must select the dictionary that will provide the values. In the **Key fields** settings block, you must use the **Add field** button to select the event fields whose values will be used for dictionary entry selection.

    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 3 elements: "a", "b" and "c". The following value is 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 two fields: the SA.StringArrayOne extended schema field and the Code field. The SA.StringArrayOne extended schema field contains 3 elements: "a", "b", and "c"; the Code string field contains the character sequence "myCode". The following value is passed to the dictionary as the key: ['a','b','c']|myCode.

  - table ?

This type of enrichment is used if you need to add a value from the dictionary of the **Table** type.

When this enrichment type is selected in the **Dictionary name** drop-down list, select the dictionary for providing the values. In the **Key fields** group of settings, use the **Add field** button to select the event fields whose values are used for dictionary entry selection.

In the **Mapping** table, configure the dictionary fields to provide data and the event fields to receive data:

- In the **Dictionary field** column, select the dictionary field. The available fields depend on the selected dictionary resource.
- In the **KUMA field** column, select the event field to which the value is written. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written to them.

New table rows can be added by using the **Add new element** button. Columns can be deleted using the ✕ button.

- event

This type of enrichment is used when you need to write a value from another event field to the current event field. Settings of this type of enrichment:

- In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.
- In the **Source field** drop-down list, select the event field whose value will be written to the target field.
- Clicking the 🔧 button opens the **Conversion** window in which you can, using the **Add conversion** button, create rules for modifying the original data before writing them to the KUMA event fields.

  Available conversions

Conversions are changes that can be applied to a value before it gets written to the event field. The conversion type is selected from a drop-down list.

Available conversions:

- **lower**—is used to make all characters of the value lowercase
- **upper**—is used to make all characters of the value uppercase
- **regexp** – used to convert a value using the regular expression RE2. When this conversion type is selected, the field appears where regular expression should be added.
- **substring**—is used to extract characters in the position range specified in the **Start** and **End** fields. These fields appear when this conversion type is selected.
- **replace**—is used to replace specified character sequence with the other character sequence. When this type of conversion is selected, new fields appear:
  - **Replace chars**—in this field you can specify the character sequence that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- **trim**—used to simultaneously remove the characters specified in the **Chars** field from the leading and end positions of the value. The field appears when this type of conversion is selected. For example, a **trim** conversion with the `Micromon` value applied to `Microsoft-Windows-Sysmon` results in `soft-Windows-Sys`.
- **append** is used to add the characters specified in the **Constant** field to the end of the event field value. The field appears when this type of conversion is selected.
- **prepend**—used to prepend the characters specified in the **Constant** field to the start of the event field value. The field appears when this type of conversion is selected.
- **replace with regexp**—is used to replace RE2 regular expression results with the character sequence.
  - **Expression**—in this field you can specify the regular expression which results that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, such a string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the "String" type, all types of conversions are available.
- For fields of the "Number" and "Float" types, the following types of conversions are available: regexp, substring, replace, trim, append, prepend, replaceWithRegexp, decodeHexString, decodeBase64String, decodeBase64URLString.
- For fields of "Array of strings", "Array of numbers", and "Array of floats" types, the following types of conversions are available: append, prepend.

When using enrichment of events that have the "Event" selected as the "Source kind" setting and the fields of the extended event schema are used as arguments, the following special considerations apply:

- If the source field is an "Array of strings" field and the target field is a "String" field, the values are written to the target field in the 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 operation is written to the DeviceCustomString1 event schema field. As a result of the operation, the DeviceCustomString1 field contains ["string1", "string2", "string3"].

- If the source field is an "Array of strings" field and the target field is an "Array of strings" field, the values are written to the target field, and commas (",") are used as the separator character.

  Example: The SA.StringArrayOne extended event schema field contains values: "string1", "string2", "string3". An event enrichment operation is performed. The result of the operation is written to the SA.StringArrayTwo event schema field. As a result of the operation, the SA.StringArrayTwo field contains "string1", "string2", "string3".

- **template**

This type of enrichment is used when you need to write a value obtained by processing Go templates into the event field. Settings of this type of enrichment:

- Put the Go template into the **Template** field.

  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.

  Example: `Attack on {{.DestinationAddress}} from {{.SourceAddress}}`.

- In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.

If you are using enrichment of events that have the "Template" type selected as the "Source kind" setting, 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.

Example:

{{.SA.StringArrayOne}}

Example:

{{- range $index, $element := . SA.StringArrayOne -}}

{{- if $index}}, {{end}}"{{$element}}"{{- end -}}

- **Target field** (required)—drop-down list for selecting the KUMA event field that should receive the data.

  This setting is not available for the enrichment source of the **Table** type.

Conditions for forwarding data to an extra normalizer

When creating additional event parsing rules, you can specify the conditions. When these conditions are met, the events are sent to the created parsing rule for processing. Conditions can be specified in the **Additional event parsing** window, on the **Extra normalization conditions** tab. This tab is not available for the basic parsing rules.

Available settings:

- **Use raw event** — If you want to send a raw event for extra normalization, select **Yes** in the **Keep raw event** drop-down list. The default value is **No**. We recommend passing a raw event to normalizers of json and xml types. If you want to send a raw event for extra normalization to the second, third, etc nesting levels, at each nesting level, select **Yes** in the **Keep raw event** drop-down list.

- **Field to pass into normalizer**—indicates the event field if you want only events with fields configured in normalizer settings to be sent for additional parsing.

  If this field is blank, the full event is sent to the extra normalizer for processing.

- Set of filters—used to define complex conditions that must be met by the events received by the normalizer.

  You can 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.

- Operators:

  - **=** – full match of the left and right operands.
  - **startsWith** – the left operand starts with the characters specified in the right operand.
  - **endsWith** – the left operand ends with the characters specified in the right operand.
  - **match** – the left operand matches the regular expression (RE2) specified in the right operand.
  - **in** – the left operand matches one of the values specified in the right operand.

The incoming data can be converted by clicking the 🔧 button. The **Conversion** window opens, where you can use 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 changes that can be applied to a value before it gets written to the event field. The conversion type is selected from a drop-down list.

Available conversions:

- **lower**—is used to make all characters of the value lowercase
- **upper**—is used to make all characters of the value uppercase
- **regexp** – used to convert a value using the regular expression RE2. When this conversion type is selected, the field appears where regular expression should be added.
- **substring**—is used to extract characters in the position range specified in the **Start** and **End** fields. These fields appear when this conversion type is selected.
- **replace**—is used to replace specified character sequence with the other character sequence. When this type of conversion is selected, new fields appear:
  - **Replace chars**—in this field you can specify the character sequence that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- **trim**—used to simultaneously remove the characters specified in the **Chars** field from the leading and end positions of the value. The field appears when this type of conversion is selected. For example, a **trim** conversion with the `Micromon` value applied to `Microsoft-Windows-Sysmon` results in `soft-Windows-Sys`.
- **append** is used to add the characters specified in the **Constant** field to the end of the event field value. The field appears when this type of conversion is selected.
- **prepend**—used to prepend the characters specified in the **Constant** field to the start of the event field value. The field appears when this type of conversion is selected.
- **replace with regexp**—is used to replace RE2 regular expression results with the character sequence.
  - **Expression**—in this field you can specify the regular expression which results that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, such a string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the "String" type, all types of conversions are available.
- For fields of the "Number" and "Float" types, the following types of conversions are available: regexp, substring, replace, trim, append, prepend, replaceWithRegexp, decodeHexString, decodeBase64String, decodeBase64URLString.
- For fields of "Array of strings", "Array of numbers", and "Array of floats" types, the following types of conversions are available: append, prepend.

Supported event sources

KUMA supports the normalization of events coming from systems listed in the "Supported event sources" table. Normalizers for these systems are included in the distribution kit.

Supported event sources

System name	Normalizer name	Type	Normalizer description
1C EventJournal	[OOTB] 1C EventJournal Normalizer	xml	Designed for processing the event log of the 1C system. The event source is the 1C log.
1C TechJournal	[OOTB] 1C TechJournal Normalizer	regexp	Designed for processing the technology event log. The event source is the 1C technology log.
Absolute Data and Device Security (DDS)	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
AhnLab Malware Defense System (MDS)	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Ahnlab UTM	[OOTB] Ahnlab UTM	regexp	Designed for processing events from the Ahnlab system. The event sources is system logs, operation logs, connections, the IPS module.
AhnLabs MDS	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Apache Cassandra	[OOTB] Apache Cassandra file	regexp	Designed for processing events from the logs of the Apache Cassandra database version 4.0.
Aruba ClearPass	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
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.

Barracuda Networks NG Firewall	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
BeyondTrust Privilege Management Console	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
BeyondTrust's BeyondInsight	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Bifit Mitigator	[OOTB] Bifit Mitigator Syslog	Syslog	Designed for processing events from the DDOS Mitigator protection system received via Syslog.
Bloombase StoreSafe	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
BMC CorreLog	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Bricata ProAccel	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Brinqa Risk Analytics	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Broadcom Symantec Advanced Threat Protection (ATP)	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
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.
Canonical LXD	[OOTB] Canonical LXD syslog	Syslog	Designed for processing events received via syslog from the Canonical LXD system version 5.18.
Checkpoint	[OOTB] Checkpoint syslog, [OOTB] Checkpoint Syslog CEF by CheckPoint	Syslog	[OOTB] Checkpoint syslog — designed for processing events received from the Checkpoint R81 firewall via the Syslog protocol. [OOTB] Checkpoint Syslog CEF by CheckPoint — designed for processing events in CEF format received from the Checkpoint firewall via the Syslog protocol.
Cisco Access Control Server (ACS)	[OOTB] Cisco ACS syslog	regexp	Designed for processing events of the Cisco Access Control Server (ACS) system received via Syslog.
Cisco ASA	[OOTB] Cisco ASA and IOS	Syslog	Designed for certain events of Cisco ASA and Cisco IOS devices received via syslog.

	syslog		
Cisco Email Security Appliance (WSA)	[OOTB] Cisco WSA AccessFile	regexp	Designed for processing the event log of the Cisco Email Security Appliance (WSA) proxy server, the access.log file.
Cisco Identity Services Engine (ISE)	[OOTB] Cisco ISE syslog	regexp	Designed for processing events of the Cisco Identity Services Engine (ISE) system received via Syslog.
Cisco IOS	[OOTB] Cisco ASA and IOS syslog	Syslog	Designed for certain events of Cisco ASA and Cisco IOS devices received via syslog.
Cisco Netflow v5	[OOTB] NetFlow v5	netflow5	Designed for processing events from Cisco Netflow version 5.
Cisco NetFlow v9	[OOTB] NetFlow v9	netflow9	Designed for processing events from Cisco Netflow version 9.
Cisco Prime	[OOTB] Cisco Prime syslog	Syslog	Designed for processing events of the Cisco Prime system version 3.10 received via syslog.
Cisco Secure Email Gateway (SEG)	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Cisco Secure Firewall Management Center	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Citrix NetScaler	[OOTB] Citrix NetScaler	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.
Corvil Network Analytics	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Cribl Stream	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
CrowdStrike Falcon Host	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
CyberArk Privileged Threat Analytics (PTA)	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
CyberPeak Spektr	[OOTB] CyberPeak Spektr syslog	Syslog	Designed for processing events of the CyberPeak Spektr system version 3 received via syslog.
DeepInstinct	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.

Delinea Secret Server	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Digital Guardian Endpoint Threat Detection	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
BIND DNS server	[OOTB] BIND Syslog [OOTB] BIND file	Syslog regexp	[OOTB] BIND Syslog is designed for processing events of the BIND DNS server received via Syslog. [OOTB] BIND file is designed for processing event logs of the BIND DNS server.
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.
EclecticIQ Intelligence Center	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Edge Technologies AppBoard and enPortal	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Eltex MES Switches	[OOTB] Eltex MES Switches	regexp	Designed for processing events from Eltex network devices.
Eset Protect	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
F5 BigIP Advanced Firewall Manager (AFM)	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
FFRI FFR yarai	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
FireEye CM Series	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
FireEye Malware Protection System	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Forcepoint NGFW	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Forcepoint SMC	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Fortinet FortiGate	[OOTB] Syslog-CEF	regexp	Designed for processing events in the CEF format.
Fortinet FortiGate	[OOTB] FortiGate syslog KV	Syslog	Designed for processing events from FortiGate firewalls via syslog. The event source is FortiGate logs in key-value format.
Fortinet Fortimail	[OOTB] Fortimail	regexp	Designed for processing events of the FortiMail email protection system. The event source is Fortimail mail system logs.
Fortinet	[OOTB] Syslog-	Syslog	Designed for processing events in the CEF format.

FortiSOAR	CEF		
FreeIPA	[OOTB] FreeIPA	json	Designed for processing events from the FreeIPA system. The event source is Free IPA directory service logs.
FreeRADIUS	[OOTB] FreeRADIUS syslog	Syslog	Designed for processing events of the FreeRADIUS system received via Syslog. The normalizer supports events from FreeRADIUS version 3.0.
Gardatech GardaDB	[OOTB] Gardatech GardaDB syslog	Syslog	Designed for processing events of the Gardatech Perimeter system version 5.3, 5.4 received via syslog.
Gardatech Perimeter	[OOTB] Gardatech Perimeter syslog	Syslog	Designed for processing events of the Gardatech Perimeter system version 5.3 received via syslog.
Gigamon GigaVUE	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
HAProxy	[OOTB] HAProxy syslog	Syslog	Designed for processing logs of the HAProxy system. The normalizer supports events of the HTTP log, TCP log, Error log type from HAProxy version 2.8.
Huawei Eudemon	[OOTB] Huawei Eudemon	regexp	Designed for processing events from Huawei Eudemon firewalls. The event source is logs of Huawei Eudemon firewalls.
Huawei USG	[OOTB] Huawei USG Basic	Syslog	Designed for processing events received from Huawei USG security gateways via Syslog.
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.
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 PAM	[OOTB] Indeed PAM syslog	Syslog	Designed for processing events of Indeed PAM (Privileged Access Manager) version 2.6.
Indeed SSO	[OOTB] Indeed SSO xml	xml	Designed for processing events of the Indeed SSO (Single Sign-On) system. The normalizer supports KUMA 2.1.3 and later.
InfoWatch 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	Designed for processing audit events received from Juniper network devices.
Kaspersky Anti Targeted Attack (KATA)	[OOTB] KATA	cef	Designed for processing alerts or events from the Kaspersky Anti Targeted Attack activity log.
Kaspersky CyberTrace	[OOTB] CyberTrace	regexp	Designed for processing Kaspersky CyberTrace events.
Kaspersky Endpoint Detection and Response (KEDR)	[OOTB] KEDR telemetry	json	Designed for processing Kaspersky EDR telemetry tagged by KATA. The event source is kafka, EnrichedEventTopic
Kaspersky Industrial CyberSecurity for Networks	[OOTB] KICS4Net v2.x	cef	Designed for processing events of Kaspersky Industrial CyberSecurity for Networks version 2.x.
Kaspersky Industrial CyberSecurity for Networks	[OOTB] KICS4Net v3.x	Syslog	Designed for processing events of Kaspersky Industrial CyberSecurity for Networks version 3.x
Kaspersky Security Center	[OOTB] KSC	cef	Designed for processing Kaspersky Security Center events received via Syslog.
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 Secure Mail Gateway (KSMG)	[OOTB] KSMG Syslog CEF	Syslog	Designed for processing events of Kaspersky Secure Mail Gateway version 2.0 in CEF format via Syslog.
Kaspersky Web Traffic Security (KWTS)	[OOTB] KWTS Syslog CEF	Syslog	Designed for processing events received from Kaspersky Web Traffic Security in CEF format via Syslog.
Kaspersky Web Traffic Security (KWTS)	[OOTB] KWTS (KV)	Syslog	Designed for processing events in Kaspersky Web Traffic Security for Key-Value format.
Kemptechnologies LoadMaster	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Kerio Control	[OOTB] Kerio Control	Syslog	Designed for processing events of Kerio Control firewalls.
KUMA	[OOTB] KUMA forwarding	json	Designed for processing events forwarded from KUMA.
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	Designed for processing events of the Linux operating system. This normalizer will be removed from the

	Syslog		OOTB set after the next release. If you are using this normalizer, you must migrate to the [OOTB] Linux audit and iptables Syslog v1 normalizer.
Linux	[OOTB] Linux audit and iptables Syslog v1	Syslog	Designed for processing events of the Linux operating system.
Linux	[OOTB] Linux audit.log file	regexp	Designed for processing security logs of Linux operating systems received via Syslog.
MariaDB	[OOTB] MariaDB Audit Plugin Syslog	Syslog	Designed for processing events coming from the MariaDB audit plugin over Syslog.
Microsoft Active Directory Federation Service (AD FS)	[OOTB] Microsoft Products	xml	Designed for processing Microsoft AD FS events. The normalizer supports this event source in KUMA 3.0.2.
Microsoft Active Directory Domain Service (AD DS)	[OOTB] Microsoft Products	xml	Designed for processing Microsoft AD DS events. The normalizer supports this event source in KUMA 3.0.2.
Microsoft Defender	[OOTB] Microsoft Products	xml	Designed for processing Microsoft Defender events.
Microsoft DHCP	[OOTB] MS DHCP file	regexp	Designed for processing Microsoft DHCP server events. The event source is Windows DHCP server logs.
Microsoft DNS	[OOTB] DNS Windows	regexp	Designed for processing Microsoft DNS server events. The event source is Windows DNS server logs.
Microsoft Exchange	[OOTB] Exchange CSV	csv	Designed for processing the event log of the Microsoft Exchange system. The event source is Exchange server MTA logs.
Microsoft IIS	[OOTB] IIS Log File Format	regexp	The normalizer processes events in the format described at https://learn.microsoft.com/en-us/windows/win32/http/iis-logging. The event source is Microsoft IIS logs.
Microsoft Network Policy Server (NPS)	[OOTB] Microsoft Products	xml	The normalizer is designed for processing events of the Microsoft Windows operating system. The event source is Network Policy Server events.
Microsoft Sysmon	[OOTB] Microsoft Products	xml	This normalizer is designed for processing Microsoft Sysmon module events.
Microsoft Windows 7, 8.1, 10, 11	[OOTB] Microsoft Products	xml	The normalizer is designed for processing some events of the Microsoft Windows operating system.
Microsoft PowerShell	[OOTB] Microsoft Products	xml	The normalizer is designed for processing events of the Microsoft Windows operating system.
Microsoft SQL Server	[OOTB] Microsoft SQL Server xml	xml	Designed for processing events of MS SQL Server versions 2008, 2012, 2014, 2016. The normalizer supports KUMA 2.1.3 and later.
Microsoft Windows Remote Desktop Services	[OOTB] Microsoft Products	xml	The normalizer is designed for processing events of the Microsoft Windows operating system. The event source is the log at Applications and Services Logs -

			Microsoft - Windows - TerminalServices-LocalSessionManager - Operational
Windows Server 2008 R2, 2012 R2, 2016, 2019, 2022	[OOTB] Microsoft Products	xml	Designed for processing some events of the Microsoft Windows Server operating system.
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.
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.
MySQL 5.7	[OOTB] MariaDB Audit Plugin Syslog	Syslog	Designed for processing events coming from the MariaDB audit plugin over Syslog.
NetApp	[OOTB] NetApp syslog, [OOTB] NetApp file	regexp	[OOTB] NetApp syslog — designed for processing events of the NetApp system (version — ONTAP 9.12) received via syslog. [OOTB] NetApp file — designed for processing events of the NetApp system (version — ONTAP 9.12) stored in a file.
NetIQ Identity Manager	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
NetScout Systems nGenius Performance Manager	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Netskope Cloud Access Security Broker	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Netwrix Auditor	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Nextcloud	[OOTB] Nextcloud syslog	Syslog	Designed for events of Nextcloud version 26.0.4 received via syslog. The normalizer does not save information from the Trace field.
Nexthink Engine	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Nginx	[OOTB] Nginx regexp	regexp	Designed for processing Nginx web server log events.
NIKSUN NetDetector	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
One Identity Privileged Session Management	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Open VPN	[OOTB] OpenVPN file	regexp	Designed for processing the event log of the OpenVPN system.
Oracle	[OOTB] Oracle	sql	Designed for processing database audit events

	Audit Trail		received by the connector directly from an Oracle database.
Orion soft zVirt	[OOTB] Orion Soft zVirt syslog	regexp	Designed for processing events of the Orion soft zVirt 3.1 virtualization system.
PagerDuty	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Palo Alto Cortex Data Lake	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Palo Alto Networks NGFW	[OOTB] PA-NGFW (Syslog-CSV)	Syslog	Designed for processing events from Palo Alto Networks firewalls received via Syslog in CSV format.
Palo Alto Networks PANOS	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
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] PTsecurity NAD	Syslog	Designed for processing events from PT Network Attack Discovery (NAD) 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	Designed for processing events from the PTsecurity (Web Application Firewall) system.
PostgreSQL pgAudit	[OOTB] PostgreSQL pgAudit Syslog	Syslog	Designed for processing events of the pgAudit audit plug-n for PostgreSQL database received via Syslog.
PowerDNS	[OOTB] PowerDNS syslog	Syslog	Designed for processing events of PowerDNS Authoritative Server 4.5 received via Syslog.
Proofpoint Insider Threat Management	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Proxmox	[OOTB] Proxmox file	regexp	Designed for processing events of the Proxmox system version 7.2-3 stored in a file. The normalizer supports processing of events in access and pveam logs.
PT NAD	[OOTB] PT NAD json	json	Designed for processing events coming from PT NAD in json format. This normalizer supports events from PT NAD version 11.1, 11.0.
QEMU - hypervisor logs	[OOTB] QEMU - Hypervisor file	regexp	Designed for processing events of the QEMU hypervisor stored in a file. QEMU 6.2.0 and Libvirt 8.0.0 are supported.
QEMU - virtual machine logs	[OOTB] QEMU - Virtual Machine file	regexp	Designed for processing events from logs of virtual machines of the QEMU hypervisor version 6.2.0, stored in a file.

Radware DefensePro AntiDDoS	[OOTB] Radware DefensePro AntiDDoS	Syslog	Designed for processing events from the DDOS Mitigator protection system received via Syslog.
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.
Recorded Future Threat Intelligence Platform	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
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.
ReversingLabs N1000 Appliance	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Rubicon Communications pfSense	[OOTB] pfSense Syslog	Syslog	Designed for processing events from the pfSense firewall received via Syslog.
Rubicon Communications pfSense	[OOTB] pfSense w/o hostname	Syslog	Designed for processing events from the pfSense firewall. The Syslog header of these events does not contain a hostname.
SailPoint IdentityIQ	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
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.
Snort	[OOTB] Snort 3 json file	json	Designed for processing events of Snort version 3 in JSON format.
Sonicwall TZ	[OOTB] Sonicwall TZ Firewall	Syslog	Designed for processing events received via Syslog from the SonicWall TZ firewall.
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
S-Terra VPN Gate	[OOTB] S-Terra	Syslog	Designed for processing events from S-Terra VPN Gate devices.
Suricata	[OOTB] Suricata json file	json	This package contains a normalizer for Suricata 7.0.1 events stored in a JSON file. The normalizer supports processing the following event types: flow, anomaly, alert, dns, http, ssl, tls, ftp, ftp_data, ftp, smb, rdp, pgsql, modbus, quic, dhcp, bittorrent_dht, rfb.

ThreatConnect Threat Intelligence Platform	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
ThreatQuotient	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
TrapX DeceptionGrid	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Trend Micro Control Manager	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Trend Micro Deep Security	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Trend Micro NGFW	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Trustwave Application Security DbProtect	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Unbound	[OOTB] Unbound Syslog	Syslog	Designed for processing events from the Unbound DNS server received via Syslog.
UserGate	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format received from the UserGate system via Syslog.
Varonis DatAdvantage	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Veriato 360	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
ViPNet TIAS	[OOTB] Vipnet TIAS syslog	Syslog	Designed for processing events of ViPNet TIAS 3.8 received via Syslog.
VMware ESXi	[OOTB] VMware ESXi syslog	regexp	Designed for processing VMware ESXi events (support for a limited number of events from ESXi versions 5.5, 6.0, 6.5, 7.0) received via Syslog.
VMWare Horizon	[OOTB] VMware Horizon - Syslog	Syslog	Designed for processing events received from the VMware Horizon 2106 system via Syslog.
VMwareCarbon Black EDR	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Vormetric Data Security Manager	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Votiro Disarmer for Windows	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Wallix AdminBastion	[OOTB] Wallix AdminBastion syslog	regexp	Designed for processing events received from the Wallix AdminBastion system via Syslog.
WatchGuard - Firebox	[OOTB] WatchGuard Firebox	Syslog	Designed for processing WatchGuard Firebox events received via Syslog.
Webroot BrightCloud	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Winchill Fracas	[OOTB] PTC	regexp	Designed for processing events of the Windchill

	Winchill Fracas		FRACAS failure registration system.
Yandex Browser corporate	[OOTB] Yandex Browser	json	Designed for processing events received from the corporate version of Yandex Browser 23.
Zabbix	[OOTB] Zabbix SQL	sql	Designed for processing events of Zabbix 6.4.
ZEEK IDS	[OOTB] ZEEK IDS json file	json	Designed for processing logs of the ZEEK IDS system in JSON format. The normalizer supports events from ZEEK IDS version 1.8.
Zettaset BDEncrypt	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Zscaler Nanolog Streaming Service (NSS)	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
IT-Bastion – SKDPU	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format received from the IT-Bastion SKDPU system via Syslog.
A-Real Internet Control Server (ICS)	[OOTB] A-real IKS syslog	regexp	Designed for processing events of the A-Real Internet Control Server (ICS) system received via Syslog. The normalizer supports events from A-Real ICS version 7.0 and later.
Apache web server	[OOTB] Apache HTTP Server file	regexp	Designed for processing Apache HTTP Server 2.4 events stored in a file. The normalizer supports processing of events from the Application log in the Common or Combined Log formats, as well as the Error log. Expected format of the Error log events: "[%t] [%-m:%l] [pid %P:tid %T] [server\ %v] [client\ %a] %E: %M;\ referer\ %-{Referer}i"
Apache web server	[OOTB] Apache HTTP Server syslog	Syslog	Designed for processing events of the Apache HTTP Server received via syslog. The normalizer supports processing of Apache HTTP Server 2.4 events from the Access log in the Common or Combined Log format, as well as the Error log. Expected format of the Error log events: "[%t] [%-m:%l] [pid %P:tid %T] [server\ %v] [client\ %a] %E: %M;\ referer\ %-{Referer}i"
Lighttpd web server	[OOTB] Lighttpd syslog	Syslog	Designed for processing Access events of the Lighttpd system received via syslog. The normalizer supports processing of Lighttpd version 1.4 events. Expected format of Access log events: $remote_addr $http_request_host_name $remote_user [$time_local] "$request" $status $body_bytes_sent "$http_referer" "$http_user_agent"
IVK Kolchuga-K	[OOTB] Kolchuga-K Syslog	Syslog	Designed for processing events from the IVK Kolchuga-K system, version LKNV.466217.002, via Syslog.
infotecs ViPNet IDS	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format received from the infotecs ViPNet IDS system via Syslog.

infotecs ViPNet Coordinator	[OOTB] VipNet Coordinator Syslog	Syslog	Designed for processing events from the ViPNet Coordinator system received via Syslog.
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	[OOTB] Confident Dallas Lock	regexp	Designed for processing events from the Dallas Lock 8 information protection system.
CryptoPro NGate	[OOTB] Ngate Syslog	Syslog	Designed for processing events received from the CryptoPro NGate system via Syslog.
NT Monitoring and Analytics	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format received from the NT Monitoring and Analytics system via Syslog.
BlueCoat proxy server	[OOTB] BlueCoat Proxy v0.2	regexp	Designed to process BlueCoat proxy server events. The event source is the BlueCoat proxy server event log.
SKDPU NT Access Gateway	[OOTB] Bastion SKDPU-GW	Syslog	Designed for processing events of the SKDPU NT Access gateway system received via Syslog.
Solar Dozor	[OOTB] Solar Dozor Syslog	Syslog	Designed for processing events received from the Solar Dozor system version 7.9 via Syslog. The normalizer supports custom format events and does not support CEF format events.
-	[OOTB] Syslog header	Syslog	Designed for processing events received via Syslog. The normalizer parses the header of the Syslog event, the message field of the event is not parsed. If necessary, you can parse the message field using other normalizers.

Aggregation rules

Aggregation rules let you combine repetitive events of the same type and replace them with one common event. Aggregation rules support fields of the standard KUMA event schema as well as fields of the extended event schema. In this way, you can reduce the number of similar events sent to the storage and/or correlator, reduce the workload on services, conserve data storage space and licensing quota (EPS). An aggregation event is created when a time or number of events threshold is reached, whichever occurs first.

For aggregation rules, you can configure a filter and apply it only to events that match the specified conditions.

You can configure aggregation rules under **Resources - Aggregation rules**, and then select the created aggregation rule from the drop-down list in the collector settings. You can also configure aggregation rules directly in the collector settings.

Available aggregation rule settings

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
Threshold	Threshold on the number of events. After accumulating the specified number of events with identical fields, the collector creates an aggregation event and begins accumulating events for the next aggregated event. The default value is `100`.
Triggered rule lifetime	Required setting. Threshold on time in seconds. When the specified time expires, the accumulation of base events stops, the collector creates an aggregated event and starts obtaining events for the next aggregated event. The default value is `60`.
Description	Resource description: up to 4,000 Unicode characters.
Identical fields	Required setting. This drop-down list lists the fields of normalized events that must have identical values. For example, for network events, you can use SourceAddress, DestinationAddress, DestinationPort fields. In the aggregation event, these fields are populated with the values of the base events.
Unique fields	This drop-down list lists the fields whose range of values must be saved in the aggregated event. For example, if the DestinationPort field is specified under **Unique fields** and not **Identical fields**, the aggregated event combines base connection events for a variety of ports, and the DestinationPort field of the aggregated event contains a list of all ports to which connections were made.
Sum fields	In this drop-down list, you can select the fields whose values will be summed up during aggregation and written to the same-name fields of the aggregated event.
Filter	Group of settings in which you can specify the conditions for identifying events that must be processed by this resource. You can select an existing filter from the drop-down list or **create** a new filter. In aggregation rules, do not use filters with the **TI** operand or the **TIDetect**, **inActiveDirectoryGroup**, or **hasVulnerability** operators. The Active Directory fields for which you can use the **inActiveDirectoryGroup** operator will appear during the enrichment stage (after aggregation rules are executed). Creating a filter in resources ?

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. The name must contain 1 to 128 Unicode characters.
4. In the **Conditions** settings block, specify the conditions that the events must meet:
   1. Click the **Add condition** button.
   2. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.
   3. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have

completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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 KUMA distribution kit includes aggregation rules listed in the table below.

Predefined aggregation rules

Aggregation rule name	Description
[OOTB] Netflow 9	The rule is triggered after 100 events or 10 seconds. Events are aggregated by fields: • DestinationAddress • DestinationPort • SourceAddress • TransportProtocol • DeviceVendor • DeviceProduct The DeviceCustomString1 and BytesIn fields are summed up.

Enrichment rules

Event enrichment involves adding information to events that can be used to identify and investigate an incident.

Enrichment rules let you add supplementary information to event fields by transforming data that is already present in the fields, or by querying data from external systems. For example, suppose that a user name is recorded in the event. You can use an enrichment rule to add information about the department, position, and manager of this user to the event fields.

Enrichment rules can be used in the following KUMA services and features:

- Collector.
- Correlator.
- Normalizer.

Available enrichment rule settings are listed in the table below.

Basic settings tab

Setting	Description
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.
Source kind	Required setting. Drop-down list for selecting the type of incoming events. Depending on the selected type, you may see the following additional settings: • **constant** This type of enrichment is used when a constant needs to be added to an event field. Settings of this type of enrichment: • In the **Constant** field, specify the value that should be added to the event field. The value may not be longer than 255 Unicode characters. If you leave this field blank, the existing event field value will be cleared. • In the **Target field** drop-down list, select the KUMA event field to which you want to write 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.

When this type is selected in the **Dictionary name** drop-down list, you must select the dictionary that will provide the values. In the **Key fields** settings block, you must use the **Add field** button to select the event fields whose values will be used for dictionary entry selection.

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 3 elements: "a", "b" and "c". The following value is 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 two fields: the SA.StringArrayOne extended schema field and the Code field. The SA.StringArrayOne extended schema field contains 3 elements: "a", "b", and "c"; the Code string field contains the character sequence "myCode". The following value is passed to the dictionary as the key: ['a','b','c']|myCode.

- table ?

This type of enrichment is used if you need to add a value from the dictionary of the **Table** type.

When this enrichment type is selected in the **Dictionary name** drop-down list, select the dictionary for providing the values. In the **Key fields** group of settings, use the **Add field** button to select the event fields whose values are used for dictionary entry selection.

In the **Mapping** table, configure the dictionary fields to provide data and the event fields to receive data:

- In the **Dictionary field** column, select the dictionary field. The available fields depend on the selected dictionary resource.
- In the **KUMA field** column, select the event field to which the value is written. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written to them.

New table rows can be added by using the **Add new element** button. Columns can be deleted using the ✕ button.

- event ?

This type of enrichment is used when you need to write a value from another event field to the current event field. Settings of this type of enrichment:

- In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.
- In the **Source field** drop-down list, select the event field whose value will be written to the target field.
- In the **Conversion** settings block, you can create rules for modifying the original data before it is written to the KUMA event fields. The conversion type can be selected from the drop-down list. You can use the **Add conversion** and **Delete** buttons to add or delete a conversion, respectively. The order of conversions is important.

  Available conversions ?

Conversions are changes that can be applied to a value before it gets written to the event field. The conversion type is selected from a drop-down list.

Available conversions:

- **lower**—is used to make all characters of the value lowercase
- **upper**—is used to make all characters of the value uppercase
- **regexp** – used to convert a value using the regular expression RE2. When this conversion type is selected, the field appears where regular expression should be added.
- **substring**—is used to extract characters in the position range specified in the **Start** and **End** fields. These fields appear when this conversion type is selected.
- **replace**—is used to replace specified character sequence with the other character sequence. When this type of conversion is selected, new fields appear:
  - **Replace chars**—in this field you can specify the character sequence that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- **trim**—used to simultaneously remove the characters specified in the **Chars** field from the leading and end positions of the value. The field appears when this type of conversion is selected. For example, a **trim** conversion with the `Micromon` value applied to `Microsoft-Windows-Sysmon` results in `soft-Windows-Sys`.
- **append** is used to add the characters specified in the **Constant** field to the end of the event field value. The field appears when this type of conversion is selected.
- **prepend**—used to prepend the characters specified in the **Constant** field to the start of the event field value. The field appears when this type of conversion is selected.
- **replace with regexp**—is used to replace RE2 regular expression results with the character sequence.
  - **Expression**—in this field you can specify the regular expression which results that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.

- **decodeBase64URLString**—used to convert a Base64url string to text.

When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, such a string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the "String" type, all types of conversions are available.
- For fields of the "Number" and "Float" types, the following types of conversions are available: regexp, substring, replace, trim, append, prepend, replaceWithRegexp, decodeHexString, decodeBase64String, decodeBase64URLString.
- For fields of "Array of strings", "Array of numbers", and "Array of floats" types, the following types of conversions are available: append, prepend.

- **template**

This type of enrichment is used when you need to write a value obtained by processing Go templates into the event field. Settings of this type of enrichment:

- Put the Go template into the **Template** field.

  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.

  Example: `Attack on {{.DestinationAddress}} from {{.SourceAddress}}`.

- In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.

If you are using enrichment of events that have the "Template" type selected as the "Source kind" setting, 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.

Example:

{{.SA.StringArrayOne}}

Example:

{{- range $index, $element := . SA.StringArrayOne -}}

{{- if $index}}, {{end}}"{{$element}}"{{- end -}}

- **dns**

This type of enrichment is used to send requests to a private network DNS server to convert IP addresses into domain names or vice versa. IP addresses are converted to DNS names only for private addresses: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 100.64.0.0/10.

Available settings:

- **URL**—in this field, you can specify the URL of a DNS server to which you want to send requests. You can use the **Add URL** button to specify multiple URLs.
- **RPS**—maximum number of requests sent to the server per second. The default value is `1,000`.
- **Workers**—maximum number of requests per one point in time. The default value is `1`.
- **Max tasks**—maximum number of simultaneously fulfilled requests. By default, this value is equal to the number of vCPUs of the KUMA Core server.
- **Cache TTL**—the lifetime of the values stored in the cache. The default value is `60`.
- **Cache disabled**—you can use this drop-down list to enable or disable caching. Caching is enabled by default.

- cybertrace ?

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.
- **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`.
- **Mapping** (required)—this settings block contains the mapping table for mapping KUMA event fields to CyberTrace indicator types. The **KUMA field** column shows the names of KUMA event fields, and the **CyberTrace indicator** column shows the types of CyberTrace indicators.

  Available types of CyberTrace indicators:

  - **ip**
  - **url**
  - **hash**

  In the mapping table, you must provide at least one string. You can use the **Add row** button to add a string, and can use the ✕ button to remove a string.

- **timezone** ?

This type of enrichment is used in collectors and correlators to assign a specific timezone to an event. Timezone information may be useful when searching for events that occurred at unusual times, such as nighttime.

When this type of enrichment is selected, the required timezone must be selected from the **Timezone** drop-down list.

> Make sure that the required time zone is set on the server hosting the enrichment-utilizing service. For example, you can do this by using the `timedatectl list-timezones` command, which shows all time zones that are set on the server. For more details on setting time zones, please refer to your operating system documentation.

When an event is enriched, the time offset of the selected timezone relative to Coordinated Universal Time (UTC) is written to the DeviceTimeZone event field in the `+-hh:mm` format. For example, if you select the **Asia/Yekaterinburg** timezone, the value `+05:00` will be written to the DeviceTimeZone field. If the enriched event already has a value in the DeviceTimeZone field, it will be overwritten.

By default, if the timezone is not specified in the event being processed and enrichment rules by timezone are not configured, the event is assigned the timezone of the server hosting the service (collector or correlator) that processes the event. If the server time is changed, the service must be restarted.

Permissible time formats when enriching the DeviceTimeZone field

When processing incoming raw events in the collector, the following time formats can be automatically converted to the +-hh:mm format:

Time format in a processed event	Example
+-hh:mm	-07:00
+-hhmm	-0700
+-hh	-07

If the date format in the `DeviceTimeZone` field differs from the formats listed above, the collector server timezone is written to the field when an event is enriched with timezone information. You can create custom normalization rules for non-standard time formats.

- **geographic data**

	This type of enrichment is used to add IP address geographic data to event fields. Learn more about linking IP addresses to geographic data. When this type is selected, in the **Mapping geographic data to event fields** settings block, you must specify from which event field the IP address will be read, select the required attributes of geographic data, and define the event fields in which geographic data will be written: 1. In the **Event field with IP address** drop-down list, select the event field from which the IP address is read. Geographic data uploaded to KUMA is matched against this IP address. You can use the **Add event field with IP address** button to specify multiple event fields with IP addresses that require geographic data enrichment. You can delete event fields added in this way by clicking the **Delete event field with IP address** button. When the `SourceAddress`, `DestinationAddress`, and `DeviceAddress` event fields are selected, the **Apply default mapping** button becomes available. You can use this button to add preconfigured mapping pairs of geographic data attributes and event fields. 2. For each event field you need to read the IP address from, select the type of geographic data and the event field to which the geographic data should be written. You can use the **Add geodata attribute** button to add field pairs for **Geodata attribute – Event field to write to**. You can also configure different types of geographic data for one IP address to be written to different event fields. To delete a field pair, click ×. - In the **Geodata attribute** field, select which geographic data corresponding to the read IP address should be written to the event. Available geographic data attributes: **Country**, **Region**, **City**, **Longitude**, **Latitude**. - In the **Event field to write to**, select the event field which the selected geographic data attribute must be written to. You can write identical geographic data attributes to different event fields. If you configure multiple geographic data attributes to be written to the same event field, the event will be enriched with the last mapping in the sequence.
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**

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. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.

- **<**—the left operand is less than the right operand.

- **<=**—the left operand is less than or equal to the right operand.

- **>**—the left operand is greater than the right operand.

- **>=**—the left operand is greater than or equal to the right operand.

- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).

- **contains**—the left operand contains values of the right operand.

- **startsWith**—the left operand starts with one of the values of the right operand.

- **endsWith**—the left operand ends with one of the values of the right operand.

- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.

- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.

- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.

- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.

- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.

- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have

completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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.

Predefined enrichment rules

The KUMA distribution kit includes enrichment rules listed in the table below.

Predefined enrichment rules

Enrichment rule name	Description
[OOTB] KATA alert	Used to enrich events received from KATA in the form of a hyperlink to an alert. The hyperlink is put in the DeviceExternalId field.

Correlation rules

Correlation rules are used to recognize specific sequences of processed events and to take certain actions after recognition, such as creating correlation events/alerts or interacting with an active list.

Correlation rules can be used in the following KUMA services and features:

- Correlator.
- Notification rule.
- Links of segmentation rules.

- Retroscan.

The available correlation rule settings depend on the selected type. Types of correlation rules:

- standard—used to find correlations between several events. Resources of this kind can create correlation events.

  This rule kind is used to determine complex correlation patterns. For simpler patterns you should use other correlation rule kinds that require less resources to operate.

- simple—used to create correlation events if a certain event is found.

- operational—used for operations with Active lists and context tables. This rule kind cannot create correlation events.

For these resources, you can enable the display of control characters in all input fields except the **Description** field.

> If a correlation rule is used in the correlator and an alert was created based on it, any change to the correlation rule will not result in a change to the existing alert even if the correlator service is restarted. For example, if the name of a correlation rule is changed, the name of the alert will remain the same. If you close the existing alert, a new alert will be created and it will take into account the changes made to the correlation rule.

Standard correlation rules

Standard correlation rules are used to identify complex patterns in processed events.

The search for patterns is conducted by using buckets ?

Bucket is a data container that is used by the Correlation rule resources to determine if the correlation event should be created. It has the following functions:

- Group together events that were matched by the filters in the **Selectors** group of settings of the Correlation rule resource. Events are grouped by the fields that were selected by user in the **Identical fields** field.
- Determine the instance when the Correlation rule should trigger, affecting the events that are grouped in the bucket.
- Perform the actions that are selected in the **Actions** group of settings.
- Create correlation events.

Available states of the Bucket:

- `Empty`—the bucket has no events. This can happen only when it was created by the correlation rule triggering.
- `Partial Match`—the bucket has some of the expected events (recovery events are not counted).
- `Full Match`—the bucket has all of the expected events (recovery events are not counted). When this condition is achieved:
  - The Correlation rule triggers
  - Events are cleared from the bucket
  - The trigger counter of the bucket is updated
  - The state of the bucket becomes `Empty`
- `False Match`—this state of the Bucket is possible:
  - when the `Full Match` state was achieved but the join-filter returned false.
  - when **Recovery** check box was selected and the recovery events were received.

  When this condition is achieved the Correlation rule does not trigger. Events are cleared from the bucket, the trigger counter is updated, and the state of the bucket becomes `Empty`.

The correlation rule window contains the following tabs:

- **General**—used to specify the main settings of the correlation rule. On this tab, you can select the type of correlation rule.
- **Selectors**—used to define the conditions that the processed events must fulfill to trigger the correlation rule. Available settings vary based on the selected rule type.
- **Actions**—used to set the triggers that will activate when the conditions configured in the **Selectors** settings block are fulfilled. The Correlation rule resource must have at least one trigger. Available settings vary based on the selected rule type.
- **Correlators**—used for linking correlators. Available only for created correlation rules that are open for editing.

General tab

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.

- **Tenant** (required)—the tenant that owns the correlation rule.

- **Type** (required)—a drop-down list for selecting the type of correlation rule. Select **standard** if you want to create a standard correlation rule.

- **Identical fields** (required)—the event fields that should be grouped in a Bucket. The hash of the values of the selected fields is used as the Bucket key. If the selector (see below) triggers, the selected fields will be copied to the correlation event.

  If different selectors of the correlation rule use fields that have different values in events, do not specify these fields in the **Identical fields** section.

- **Unique fields**—event fields that should be sent to the Bucket. If this parameter is set, the Bucket will receive only unique events. The hash of the selected fields' values is used as the Bucket key.

> You can use local variables in the **Identical fields** and **Unique fields** sections. To access a variable, its name must be preceded with the "$" character.
> For an example of using local variables in these sections, refer to the rule provided with KUMA: R403_Access to malicious resources from a host with disabled protection or an out-of-date anti-virus database.

- **Rate limit**—maximum number of times a correlation rule can be triggered per second. The default value is 100.

> If correlation rules employing complex logic for pattern detection are not triggered, this may be due to the specific method used to count rule triggers in KUMA. In this case, try to increase the value of **Rate limit** to `1000000`, for example.

- **Window, sec** (required)—bucket lifetime, in seconds. Default value: 86,400 seconds (24 hours). This timer starts when the Bucket is created (when it receives the first event). The lifetime is not updated, and when it runs out, the **On timeout** trigger from the **Actions** group of settings is activated and the bucket is deleted. The **On every threshold** and **On subsequent thresholds** triggers can be activated more than once during the lifetime of the bucket.

- **Base events keep policy**—this drop-down list is used to specify which base events must be stored in the correlation event:

  - **first** (default value)—this option is used to store the first base event of the event collection that triggered creation of the correlation event.

  - **last**—this option is used to store the last base event of the event collection that triggered creation of the correlation event.

  - **all**—this option is used to store all base events of the event collection that triggered creation of the correlation event.

- **Priority**—base coefficient used to determine the importance of a correlation rule. The default value is **Low**.

- **Order by**—in this drop-down list, you can select the event field that will be used by the correlation rule selectors to track situational changes. This could be useful if you want to configure a correlation rule to be triggered when several types of events occur sequentially, for example.

- **Description**—the description of a resource. Up to 4,000 Unicode characters.

Selectors tab

A rule of the **standard** kind can have multiple selectors. You can add selectors by clicking the **Add selector** button and can remove them by clicking the **Delete selector** button. Selectors can be moved by using the ⠿ button.

For each selector, the following two tabs are available: **Settings** and **Local variables**.

The **Settings** tab contains the following settings:

- **Alias** (required)—unique name of the event group that meets the conditions of the selector. Must contain 1 to 128 Unicode characters.

- **Selector threshold (event count)** (required)—the number of events that must be received by the selector to trigger. The default value is 1.

- **Filter** (required)—used to set the criteria for determining events that should trigger the selector. You can select an existing filter from the drop-down list or **create** a new filter.

  Creating a filter in resources

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. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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`.

    A value of this type is specified 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.

    A value of this type is specified manually.

    To work with a value from the Extra field at depth 3 and below, use backquotes ``. For example, `Extra.lev1.lev2.lev3`.
- Operator – =.
- Right operand—**constant**.
- Value—the value by which you need to filter events.

The order of conditions specified in the selector filter of the correlation rule is significant and affects system performance. We recommend putting the most unique condition in the first place in the selector filter.

Consider two examples of selector filters that select successful authentication events in Microsoft Windows.

Selector filter 1:

Condition 1. DeviceProduct = Microsoft Windows

Condition 2. DeviceEventClassID = 4624

Selector filter 2:

Condition 1. DeviceEventClassID = 4624

Condition 2. DeviceProduct = Microsoft Windows

The order of conditions in Selector filter 2 is preferable because it causes less load on the system.

- **Recovery**—this check box must be selected when the Correlation rule must NOT trigger if a certain number of events are received from the selector. By default, this check box is cleared.

Select the **Local variables** tab and click **Add variable** to declare variables that you want to use within the limits of this correlation rule.

In the selector of the correlation rule, you can use regular expressions conforming to the RE2 standard.

Using regular expressions in correlation rules is computationally intensive compared to other operations. Therefore, when designing correlation rules, we recommend limiting the use of regular expressions to the necessary minimum and using other available operations.

To use a regular expression, you must use the `match` comparison operator. The regular expression must be placed in a constant. The use of capture groups in regular expressions is optional. For the correlation rule to trigger, the field text matched against the regexp must exactly match the regular expression.

For a primer on syntax and examples of correlation rules that use regular expressions in their selectors, see the following rules that are provided with KUMA:

- R105_04_Suspicious PowerShell commands. Suspected obfuscation.
- R333_Suspicious creation of files in the autorun folder.

Actions tab

A rule of the **standard** kind can have multiple triggers.

- **On first threshold**—this trigger activates when the Bucket registers the first triggering of the selector during the lifetime of the Bucket.
- **On subsequent thresholds**—this trigger activates when the Bucket registers the second and all subsequent triggering of the selector during the lifetime of the Bucket.
- **On every threshold**—this trigger activates every time the Bucket registers the triggering of the selector.
- **On timeout**—this trigger activates when the lifetime of the Bucket ends, and is linked to the selector with the **Recovery** check box selected. In other words, this trigger activates if the situation detected by the correlation rule is not resolved within the defined amount of time.

Every trigger is represented as a group of settings with the following parameters available:

- **Output**—if this check box is selected, the correlation event is sent for post-processing: for external enrichment outside the correlation rule, for response, and to destinations.
- **Loop to correlator**—if this check box is selected, the created correlation event is processed by the rule chain of the current correlator. This allows hierarchical correlation.

  > If both check boxes are selected, the correlation rule will be sent for post-processing first and then to the current correlation rule selectors.

- **Do not create alert**—if this check box is selected, no alert is created when this correlation rule is triggered.
- Under **Enrichment**, you can modify the fields of correlation events by using enrichment rules. These enrichment rules are stored in the correlation rule where they were created. You can create multiple enrichment rules. Enrichment rules can be added or deleted by using the **Add enrichment** or **Remove enrichment** buttons, respectively.
  - **Source kind**—you can select the type of enrichment in this drop-down list. Depending on the selected type, you may see advanced settings that will also need to be completed.

    Available types of enrichment:

    - constant ?

      > This type of enrichment is used when a constant needs to be added to an event field. Settings of this type of enrichment:
      >
      > - In the **Constant** field, specify the value that should be added to the event field. The value may not be longer than 255 Unicode characters. If you leave this field blank, the existing event field value will be cleared.
      > - In the **Target field** drop-down list, select the KUMA event field to which you want to write 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.

When this type is selected in the **Dictionary name** drop-down list, you must select the dictionary that will provide the values. In the **Key fields** settings block, you must use the **Add field** button to select the event fields whose values will be used for dictionary entry selection.

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 3 elements: "a", "b" and "c". The following value is 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 two fields: the SA.StringArrayOne extended schema field and the Code field. The SA.StringArrayOne extended schema field contains 3 elements: "a", "b", and "c"; the Code string field contains the character sequence "myCode". The following value is passed to the dictionary as the key: ['a','b','c']|myCode.

- table ?

This type of enrichment is used if you need to add a value from the dictionary of the **Table** type.

When this enrichment type is selected in the **Dictionary name** drop-down list, select the dictionary for providing the values. In the **Key fields** group of settings, use the **Add field** button to select the event fields whose values are used for dictionary entry selection.

In the **Mapping** table, configure the dictionary fields to provide data and the event fields to receive data:

- In the **Dictionary field** column, select the dictionary field. The available fields depend on the selected dictionary resource.
- In the **KUMA field** column, select the event field to which the value is written. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written to them.

New table rows can be added by using the **Add new element** button. Columns can be deleted using the ✕ button.

- event ?

This type of enrichment is used when you need to write a value from another event field to the current event field. Settings of this type of enrichment:

- In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.
- In the **Source field** drop-down list, select the event field whose value will be written to the target field.
- Clicking the 🔧 button opens the **Conversion** window in which you can, using the **Add conversion** button, create rules for modifying the original data before writing them to the KUMA event fields.

  Available conversions

Conversions are changes that can be applied to a value before it gets written to the event field. The conversion type is selected from a drop-down list.

Available conversions:

- **lower**—is used to make all characters of the value lowercase
- **upper**—is used to make all characters of the value uppercase
- **regexp** – used to convert a value using the regular expression RE2. When this conversion type is selected, the field appears where regular expression should be added.
- **substring**—is used to extract characters in the position range specified in the **Start** and **End** fields. These fields appear when this conversion type is selected.
- **replace**—is used to replace specified character sequence with the other character sequence. When this type of conversion is selected, new fields appear:
  - **Replace chars**—in this field you can specify the character sequence that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- **trim**—used to simultaneously remove the characters specified in the **Chars** field from the leading and end positions of the value. The field appears when this type of conversion is selected. For example, a **trim** conversion with the `Micromon` value applied to `Microsoft-Windows-Sysmon` results in `soft-Windows-Sys`.
- **append** is used to add the characters specified in the **Constant** field to the end of the event field value. The field appears when this type of conversion is selected.
- **prepend**—used to prepend the characters specified in the **Constant** field to the start of the event field value. The field appears when this type of conversion is selected.
- **replace with regexp**—is used to replace RE2 regular expression results with the character sequence.
  - **Expression**—in this field you can specify the regular expression which results that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, such a string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the "String" type, all types of conversions are available.
- For fields of the "Number" and "Float" types, the following types of conversions are available: regexp, substring, replace, trim, append, prepend, replaceWithRegexp, decodeHexString, decodeBase64String, decodeBase64URLString.
- For fields of "Array of strings", "Array of numbers", and "Array of floats" types, the following types of conversions are available: append, prepend.

When using enrichment of events that have the "Event" selected as the "Source kind" setting and the fields of the extended event schema are used as arguments, the following special considerations apply:

- If the source field is an "Array of strings" field and the target field is a "String" field, the values are written to the target field in the 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 operation is written to the DeviceCustomString1 event schema field. As a result of the operation, the DeviceCustomString1 field contains ["string1", "string2", "string3"].

- If the source field is an "Array of strings" field and the target field is an "Array of strings" field, the values are written to the target field, and commas (",") are used as the separator character.

  Example: The SA.StringArrayOne extended event schema field contains values: "string1", "string2", "string3". An event enrichment operation is performed. The result of the operation is written to the SA.StringArrayTwo event schema field. As a result of the operation, the SA.StringArrayTwo field contains "string1", "string2", "string3".

- **template**

This type of enrichment is used when you need to write a value obtained by processing Go templates into the event field. Settings of this type of enrichment:

- Put the Go template into the **Template** field.

  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.

  Example: `Attack on {{.DestinationAddress}} from {{.SourceAddress}}`.

- In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.

If you are using enrichment of events that have the "Template" type selected as the "Source kind" setting, 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.

Example:

{{.SA.StringArrayOne}}

Example:

{{- range $index, $element := . SA.StringArrayOne -}}

{{- if $index}}, {{end}}"{{$element}}"{{- end -}}

  - **Debug**—you can use this toggle switch to enable logging of service operations.

  - **Description**—the description of a resource. Up to 4,000 Unicode characters.

- **Categorization** settings group—used to change the categories of assets indicated in events. There can be several categorization rules. You can add or delete them by using the **Add categorization** or **Remove categorization** buttons. Only reactive categories can be added to assets or removed from assets.

  - **Operation**—this drop-down list is used to select the operation to perform on the category:

    - **Add**—assign the category to the asset.

    - **Delete**—unbind the asset from the category.

  - **Event field**—event field that indicates the asset requiring the operation.

  - **Category ID**—the drop-down list displays a tree of categories, in which you can select a category to perform the operation on. Clicking the row expands the list.

- **Active lists update** settings group—used to assign the trigger for one or more operations with active lists. You can use the **Add active list action** and **Delete active list action** buttons to add or delete operations with active lists, respectively.

  Available settings:

  - **Name** (required)—this drop-down list is used to select the Active list resources.

  - **Operation** (required)—this drop-down list is used to select the operation that must be performed:

        - **Sum**—add a constant, the value of a correlation event field, or the value of a local variable to the value of the active list.
        - **Get**—get the Active list entry and write the values of the selected fields into the correlation event.
        - **Set**—write the values of the selected fields of the correlation event into the Active list by creating a new or updating an existing Active list entry. When the Active list entry is updated, the data is merged and only the specified fields are overwritten.
        - **Delete**—delete the Active list entry.
    - **Key fields** (required)—this is the list of event fields used to create the Active list entry. It is also used as the Active list entry key.

      > The active list entry key depends on the available fields and does not depend on the order in which they are displayed in the KUMA web interface.

    - **Mapping** (required for **Get** and **Set** operations)—used to map Active list fields with events fields. More than one mapping rule can be set.
        - The left field is used to specify the Active list field.

          > The field must not contain special characters or numbers only.

        - The middle drop-down list is used to select event fields.
        - The right field can be used to assign a constant to the Active list field is the **Set** operation was selected.
- Under **Context table update**, you can assign the trigger for one or more operations with context tables. You can click **Add context table action** or **Delete context table action** to add or delete operations with context tables, respectively.

  Available settings:

    - **Name** (required)—this drop-down list is used to select context table resources.
    - **Operation** (required)—this drop-down list is used to select the operation that must be performed.
        - **Sum**—add a constant, the value of a correlation event field, or the value of a local variable to the value of the context table. This operation is used only for fields of number and float types.
        - **Set**—write the values of the selected fields of the correlation event into the context table by creating a new or updating an existing context table entry. When the context table entry is updated, the data is merged and only the specified fields are overwritten.
        - **Get**—get the fields of the context table and write the values of the specified fields into the correlation event. Table fields of the boolean type and lists of boolean values are excluded from mapping because the event does not contain boolean fields.
        - **Merge**—append the value of a correlation event field, local variable, or constant to the current value of a field of the context table.
        - **Delete**—delete the context table entry.

- **Key fields** (required)—this is the list of event fields used to create the context table entry. It is also used as the key of the context table entry. As a key field, you can specify an event field or a local variable declared on the**Selectors** **tab**.

  > The composite key of the context table entry depends only on the values of fields and does not depend on the order in which they are displayed in the KUMA web interface.

- **Mapping** (required for all operations except **Delete**)—used to map context table fields to event fields or variables. More than one mapping rule can be set. You can specify the same context table field multiple times.
  - The left field is used to specify the context table field.

    The field must not contain a field name that is already used in the mapping, tab characters, special characters, or only numerals. The maximum number of characters is 128. The name cannot begin with an underscore.
  - The middle drop-down list is used to select event fields or a local variable.
  - The right field can be used to assign a constant to the context table field is the **Set** operation was selected. **Merge** or **Sum**. The maximum number of characters is 1,024.

Correlators tab

- **Add**—Used when editing the created correlation rule. You can click **Add** to open the **Correlators** window and select a correlator from the list. After you click **OK**, the rule is linked to the selected correlator. You can select multiple correlators at the same time. The rule is added to the end of the execution queue. If you want to move the rule up in the execution queue, go to **Resources** - **Correlator** - <selected correlator> - **Edit correlator** - **Correlation**, select the check box next to the relevant rule and use the **Move up** or **Move down** buttons to reorder the rules as necessary.
- **Delete**—Used to unlink the correlation rule from the correlator.

Simple correlation rules

Simple correlation rules are used to define simple sequences of events.

The correlation rule window contains the following settings tabs:

- **General**—used to specify the main settings of the correlation rule. On this tab, you can select the type of correlation rule.
- **Selectors**—used to define the conditions that the processed events must fulfill to trigger the correlation rule. Available settings vary based on the selected rule kind.
- **Actions**—used to set the triggers that will activate when the conditions configured in the **Selectors** settings block are fulfilled. A correlation rule must have at least one trigger. Available settings vary based on the selected rule type.
- **Correlators**—used for linking correlators. Available only for created correlation rules that are open for editing.

General tab

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—the tenant that owns the correlation rule.
- **Type** (required)—a drop-down list for selecting the type of correlation rule. Select **simple** if you want to create a simple correlation rule.
- **Propagated fields** (required)—event fields used for event selection. If the selector (see below) is triggered, these fields will be written to the correlation event.
- **Rate limit**—maximum number of times a correlation rule can be triggered per second. The default value is 100.

  > If correlation rules employing complex logic for pattern detection are not triggered, this may be due to the specific method used to count rule triggers in KUMA. In this case, try to increase the value of **Rate limit** to `1000000`, for example.

- **Priority**—base coefficient used to determine the importance of a correlation rule. The default value is `Low`.
- **Description**—the description of a resource. Up to 4,000 Unicode characters.

Selectors tab

A rule of the **simple** kind can have only one selector for which the **Settings** and **Local variables** tabs are available.

The **Settings** tab contains settings with the **Filter** group of settings:

- **Filter** (required)—used to set the criteria for determining events that should trigger the selector. You can select an existing filter from the drop-down list or **create** a new filter.

  Creating a filter in resources ?

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. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

   The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

   This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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`.

    A value of this type is specified 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.

    A value of this type is specified manually.

    To work with a value from the Extra field at depth 3 and below, use backquotes ``. For example, `Extra.lev1.lev2.lev3`.
- Operator – =.
- Right operand—**constant**.
- Value—the value by which you need to filter events.

The order of conditions specified in the selector filter of the correlation rule is significant and affects system performance. We recommend putting the most unique condition in the first place in the selector filter.

Consider two examples of selector filters that select successful authentication events in Microsoft Windows.

Selector filter 1:

Condition 1. DeviceProduct = Microsoft Windows

Condition 2. DeviceEventClassID = 4624

Selector filter 2:

Condition 1. DeviceEventClassID = 4624

Condition 2. DeviceProduct = Microsoft Windows

The order of conditions in Selector filter 2 is preferable because it causes less load on the system.

Select the **Local variables** tab and click **Add variable** to declare variables that you want to use within the limits of this correlation rule.

Actions tab

A rule of the **simple** kind can have only one trigger: **On every event**. It is activated every time the selector triggers.

Available parameters of the trigger:

- **Output**—if this check box is selected, the correlation event is sent for post-processing: for enrichment, for a response, and to destinations.

- **Loop to correlator**—if this check box is selected, the correlation event will be processed by the current correlation rule. This allows hierarchical correlation.

  > If both check boxes are selected, the correlation rule will be sent for post-processing first and then to the current correlation rule selectors.

- **Do not create alert**—if this check box is selected, no alert is created when this correlation rule is triggered.

- **Enrichment** settings group—you can modify the fields of correlation events by using enrichment rules. These enrichment rules are stored in the correlation rule where they were created. You can create multiple enrichment rules. Enrichment rules can be added or deleted by using the **Add enrichment** or **Remove enrichment** buttons, respectively.

  - **Source kind**—you can select the type of enrichment in this drop-down list. Depending on the selected type, you may see advanced settings that will also need to be completed.

    Available types of enrichment:

    - constant

This type of enrichment is used when a constant needs to be added to an event field. Settings of this type of enrichment:

- In the **Constant** field, specify the value that should be added to the event field. The value may not be longer than 255 Unicode characters. If you leave this field blank, the existing event field value will be cleared.
- In the **Target field** drop-down list, select the KUMA event field to which you want to write 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.

When this type is selected in the **Dictionary name** drop-down list, you must select the dictionary that will provide the values. In the **Key fields** settings block, you must use the **Add field** button to select the event fields whose values will be used for dictionary entry selection.

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 3 elements: "a", "b" and "c". The following value is 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 two fields: the SA.StringArrayOne extended schema field and the Code field. The SA.StringArrayOne extended schema field contains 3 elements: "a", "b", and "c"; the Code string field contains the character sequence "myCode". The following value is passed to the dictionary as the key: ['a','b','c']|myCode.

- table

This type of enrichment is used if you need to add a value from the dictionary of the **Table** type.

When this enrichment type is selected in the **Dictionary name** drop-down list, select the dictionary for providing the values. In the **Key fields** group of settings, use the **Add field** button to select the event fields whose values are used for dictionary entry selection.

In the **Mapping** table, configure the dictionary fields to provide data and the event fields to receive data:

- In the **Dictionary field** column, select the dictionary field. The available fields depend on the selected dictionary resource.
- In the **KUMA field** column, select the event field to which the value is written. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written to them.

New table rows can be added by using the **Add new element** button. Columns can be deleted using the ✕ button.

- event

This type of enrichment is used when you need to write a value from another event field to the current event field. Settings of this type of enrichment:

- In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.
- In the **Source field** drop-down list, select the event field whose value will be written to the target field.
- Clicking the 🔧 button opens the **Conversion** window in which you can, using the **Add conversion** button, create rules for modifying the original data before writing them to the KUMA event fields.

  Available conversions

Conversions are changes that can be applied to a value before it gets written to the event field. The conversion type is selected from a drop-down list.

Available conversions:

- **lower**—is used to make all characters of the value lowercase
- **upper**—is used to make all characters of the value uppercase
- **regexp** – used to convert a value using the regular expression RE2. When this conversion type is selected, the field appears where regular expression should be added.
- **substring**—is used to extract characters in the position range specified in the **Start** and **End** fields. These fields appear when this conversion type is selected.
- **replace**—is used to replace specified character sequence with the other character sequence. When this type of conversion is selected, new fields appear:
  - **Replace chars**—in this field you can specify the character sequence that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- **trim**—used to simultaneously remove the characters specified in the **Chars** field from the leading and end positions of the value. The field appears when this type of conversion is selected. For example, a **trim** conversion with the `Micromon` value applied to `Microsoft-Windows-Sysmon` results in `soft-Windows-Sys`.
- **append** is used to add the characters specified in the **Constant** field to the end of the event field value. The field appears when this type of conversion is selected.
- **prepend**—used to prepend the characters specified in the **Constant** field to the start of the event field value. The field appears when this type of conversion is selected.
- **replace with regexp**—is used to replace RE2 regular expression results with the character sequence.
  - **Expression**—in this field you can specify the regular expression which results that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, such a string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the "String" type, all types of conversions are available.
- For fields of the "Number" and "Float" types, the following types of conversions are available: regexp, substring, replace, trim, append, prepend, replaceWithRegexp, decodeHexString, decodeBase64String, decodeBase64URLString.
- For fields of "Array of strings", "Array of numbers", and "Array of floats" types, the following types of conversions are available: append, prepend.

When using enrichment of events that have the "Event" selected as the "Source kind" setting and the fields of the extended event schema are used as arguments, the following special considerations apply:

- If the source field is an "Array of strings" field and the target field is a "String" field, the values are written to the target field in the 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 operation is written to the DeviceCustomString1 event schema field. As a result of the operation, the DeviceCustomString1 field contains ["string1", "string2", "string3"].

- If the source field is an "Array of strings" field and the target field is an "Array of strings" field, the values are written to the target field, and commas (",") are used as the separator character.

  Example: The SA.StringArrayOne extended event schema field contains values: "string1", "string2", "string3". An event enrichment operation is performed. The result of the operation is written to the SA.StringArrayTwo event schema field. As a result of the operation, the SA.StringArrayTwo field contains "string1", "string2", "string3".

- **template** ?

This type of enrichment is used when you need to write a value obtained by processing Go templates into the event field. Settings of this type of enrichment:

- Put the Go template into the **Template** field.

  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.

  Example: `Attack on {{.DestinationAddress}} from {{.SourceAddress}}`.

- In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.

If you are using enrichment of events that have the "Template" type selected as the "Source kind" setting, 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.

Example:

{{.SA.StringArrayOne}}

Example:

{{- range $index, $element := . SA.StringArrayOne -}}

{{- if $index}}, {{end}}"{{$element}}"{{- end -}}

- **Debug**—you can use this toggle switch to enable logging of service operations.

- **Description**—the description of a resource. Up to 4,000 Unicode characters.

- **Filter** settings block—lets you select which events will be forwarded for enrichment. Configuration is performed as described above.

- **Categorization** settings group—used to change the categories of assets indicated in events. There can be several categorization rules. You can add or delete them by using the **Add categorization** or **Remove categorization** buttons. Only reactive categories can be added to assets or removed from assets.

  - **Operation**—this drop-down list is used to select the operation to perform on the category:

    - **Add**—assign the category to the asset.

    - **Delete**—unbind the asset from the category.

  - **Event field**—event field that indicates the asset requiring the operation.

  - **Category ID**—the drop-down list displays a tree of categories, in which you can select a category to perform the operation on. Clicking the row expands the list.

- **Active lists update** settings group—used to assign the trigger for one or more operations with active lists. You can use the **Add active list action** and **Delete active list action** buttons to add or delete operations with active lists, respectively.

  Available settings:

- **Name** (required)—this drop-down list is used to select the active list.
- **Operation** (required)—this drop-down list is used to select the operation that must be performed:
    - **Sum**—add a constant, the value of a correlation event field, or the value of a local variable to the value of the active list.
    - **Get**—get the Active list entry and write the values of the selected fields into the correlation event.
    - **Set**—write the values of the selected fields of the correlation event into the Active list by creating a new or updating an existing Active list entry. When the Active list entry is updated, the data is merged and only the specified fields are overwritten.
    - **Get**—get the Active list entry and write the values of the selected fields into the correlation event.
    - **Delete**—delete the Active list entry.
- **Key fields** (required)—this is the list of event fields used to create the Active list entry. It is also used as the Active list entry key.

    > The active list entry key depends on the available fields and does not depend on the order in which they are displayed in the KUMA web interface.

- **Mapping** (required for **Get** and **Set** operations)—used to map Active list fields with events fields. More than one mapping rule can be set.
    - The left field is used to specify the Active list field.

        > The field must not contain special characters or numbers only.

    - The middle drop-down list is used to select event fields.
    - The right field can be used to assign a constant to the Active list field is the **Set** operation was selected.

- Under **Context table update**, you can assign the trigger for one or more operations with context tables. You can click **Add context table action** or **Delete context table action** to add or delete operations with context tables, respectively.

  Available settings:

    - **Name** (required)—this drop-down list is used to select context table resources.
    - **Operation** (required)—this drop-down list is used to select the operation that must be performed.
        - **Sum**—add a constant, the value of a correlation event field, or the value of a local variable to the value of the context table. This operation is used only for fields of number and float types.
        - **Set**—write the values of the selected fields of the correlation event into the context table by creating a new or updating an existing context table entry. When the context table entry is updated, the data is merged and only the specified fields are overwritten.
        - **Get**—get the fields of the context table and write the values of the specified fields into the correlation event. Table fields of the boolean type and lists of boolean values are excluded from mapping because the event does not contain boolean fields.

  - **Merge**—append the value of a correlation event field, local variable, or constant to the current value of a field of the context table.
  - **Delete**—delete the context table entry.
- **Key fields** (required)—this is the list of event fields used to create the context table entry. It is also used as the key of the context table entry. As a key field, you can specify an event field or a local variable declared on the**Selectors** **tab**.

  > The composite key of the context table entry depends only on the values of fields and does not depend on the order in which they are displayed in the KUMA web interface.

- **Mapping** (required for all operations except **Delete**)—used to map context table fields to event fields or variables. More than one mapping rule can be set. You can specify the same context table field multiple times.
  - The left field is used to specify the context table field.

    The field must not contain a field name that is already used in the mapping, tab characters, special characters, or only numerals. The maximum number of characters is 128. The name cannot begin with an underscore.
  - The middle drop-down list is used to select event fields or a local variable.
- The right field can be used to assign a constant to the context table field is the **Set** operation was selected. **Merge** or **Sum**. The maximum number of characters is 1,024.

Correlators tab

- **Add**—Used when editing the created correlation rule. You can click **Add** to open the **Correlators** window and select a correlator from the list. After you click **OK**, the rule is linked to the selected correlator. You can select multiple correlators at the same time. The rule is added to the end of the execution queue. If you want to move the rule up in the execution queue, go to **Resources** - **Correlator** - <selected correlator> - **Edit correlator** - **Correlation**, select the check box next to the relevant rule and use the Move up or Move down buttons to reorder the rules as necessary.
- **Delete**—Used to unlink the correlation rule from the correlator.

Operational correlation rules

Operational correlation rules are used for working with active lists.

The correlation rule window contains the following tabs:

- **General**—used to specify the main settings of the correlation rule. On this tab, you can select the type of correlation rule.
- **Selectors**—used to define the conditions that the processed events must fulfill to trigger the correlation rule. Available settings vary based on the selected rule type.
- **Actions**—used to set the triggers that will activate when the conditions configured in the **Selectors** settings block are fulfilled. A correlation rule must have at least one trigger. Available settings vary based on the

selected rule type.

- **Correlators**—used for linking correlators. Available only for created correlation rules that are open for editing.

General tab

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—the tenant that owns the correlation rule.
- **Type** (required)—a drop-down list for selecting the type of correlation rule. Select **operational** if you want to create an operational correlation rule.
- **Rate limit**—maximum number of times a correlation rule can be triggered per second. The default value is 100.

  > If correlation rules employing complex logic for pattern detection are not triggered, this may be due to the specific method used to count rule triggers in KUMA. In this case, try to increase the value of **Rate limit** to `1000000`, for example.

- **Description**—the description of a resource. Up to 4,000 Unicode characters.

Selectors tab

A rule of the **operational** kind can have only one selector for which the **Settings** and **Local variables** tabs are available.

The **Settings** tab contains settings with the **Filter** group of settings:

- **Filter** (required)—used to set the criteria for determining events that should trigger the selector. You can select an existing filter from the drop-down list or **create** a new filter.

  Creating a filter in resources

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. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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`.

    A value of this type is specified 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.

    A value of this type is specified manually.

    To work with a value from the Extra field at depth 3 and below, use backquotes ``. For example, `Extra.lev1.lev2.lev3`.

- Operator – =.

- Right operand—**constant**.

- Value—the value by which you need to filter events.

On the **Local variables** tab, click **Add variable** to declare variables that you want to use within the limits of this correlation rule.

Actions tab

A rule of the **operational** kind can have only one trigger: **On every event**. It is activated every time the selector triggers.

Available parameters of the trigger:

- **Active lists update** settings group—used to assign the trigger for one or more operations with active lists. You can use the **Add active list action** and **Delete active list action** buttons to add or delete operations with active lists, respectively.

  Available settings:

  - **Name** (required)—this drop-down list is used to select the active list.
  - **Operation** (required)—this drop-down list is used to select the operation that must be performed:
    - **Sum**—add a constant, the value of a correlation event field, or the value of a local variable to the value of the active list.
    - **Set**—write the values of the selected fields of the correlation event into the Active list by creating a new or updating an existing Active list entry. When the Active list entry is updated, the data is merged and only the specified fields are overwritten.
    - **Delete**—delete the Active list entry.
  - **Key fields** (required)—this is the list of event fields used to create the Active list entry. It is also used as the Active list entry key.

    > The active list entry key depends on the available fields and does not depend on the order in which they are displayed in the KUMA web interface.

  - **Mapping** (required for **Set** operation)—used to map active list fields with event fields. More than one mapping rule can be set.
    - The left field is used to specify the Active list field.

      > The field must not contain special characters or numbers only.

    - The middle drop-down list is used to select event fields.
    - The right field can be used to assign a constant to the Active list field is the **Set** operation was selected.

- Under **Context table update**, you can assign the trigger for one or more operations with context tables. You can click **Add context table action** or **Delete context table action** to add or delete operations with context tables, respectively.

  Available settings:

  - **Name** (required)—this drop-down list is used to select context table resources.

- **Operation** (required)—this drop-down list is used to select the operation that must be performed.
  - **Sum**—add a constant, the value of a correlation event field, or the value of a local variable to the value of the context table. This operation is used only for fields of number and float types.
  - **Set**—write the values of the selected fields of the correlation event into the context table by creating a new or updating an existing context table entry. When the context table entry is updated, the data is merged and only the specified fields are overwritten.
  - **Merge**—append the value of a correlation event field, local variable, or constant to the current value of a field of the context table.
  - **Delete**—delete the context table entry.
- **Key fields** (required)—this is the list of event fields used to create the context table entry. It is also used as the key of the context table entry. As a key field, you can specify an event field or a local variable declared on the**Selectors** **tab**.

  > The composite key of the context table entry depends only on the values of fields and does not depend on the order in which they are displayed in the KUMA web interface.

- **Mapping** (required for all operations except **Delete**)—used to map context table fields to event fields or variables. More than one mapping rule can be set. You can specify the same context table field multiple times.
  - The left field is used to specify the context table field.

    The field must not contain a field name that is already used in the mapping, tab characters, special characters, or only numerals. The maximum number of characters is 128. The name cannot begin with an underscore.
  - The middle drop-down list is used to select event fields or a local variable.
  - You can use the right field to assign a constant to the context table field. The maximum number of characters is 1024.

Correlators tab

- **Add**—Used when editing the created correlation rule. You can click **Add** to open the **Correlators** window and select a correlator from the list. After you click **OK**, the rule is linked to the selected correlator. You can select multiple correlators at the same time. The rule is added to the end of the execution queue. If you want to move the rule up in the execution queue, go to **Resources** - **Correlator** - <selected correlator> - **Edit correlator** - **Correlation**, select the check box next to the relevant rule and use the Move up or Move down buttons to reorder the rules as necessary.
- **Delete**—Used to unlink the correlation rule from the correlator.

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" functions

These functions allow you to receive information from an active list and dynamically generate a field name for an active list and key.

You must specify the parameters in the following sequence:

1. Name of the active list

2. Expression that returns the field name of the active list

3. One or more expressions whose results are used to generate the key

Usage example	Result
`active_list('Test', to_lower('DeviceHostName'), to_lower(DeviceCustomString2), to_lower(DeviceCustomString1))`	Gets the field value of the active list.

Use these functions to query the active list of the shared tenant from a variable. To do so, add the @Shared suffix after the name of the active list (case sensitive). For example, active_list('exampleActiveList@Shared', 'score', SourceAddress, SourceUserName).

"table_dict" function

Gets information about the value in the specified column of a dictionary of the table type.

You must specify the parameters in the following sequence:

1. Dictionary name

2. Dictionary column name

3. One or more expressions whose results are used to generate the dictionary row key.

Usage example	Result
`table_dict('exampleTableDict', 'office', SourceUserName)`	Gets data from the `exampleTableDict` dictionary from the row with the `SourceUserName` key in the `office` column.
`table_dict('exampleTableDict', 'office', SourceAddress, to_lower(SourceUserName))`	Gets data from the `exampleTableDict` dictionary from a composite key string from the `SourceAddress` field value and the lowercase value of the `SourceUserName` field from the `office` column.

Use this function to access the dictionary of the shared tenant from a variable. To do so, add the `@Shared` suffix after the name of the active list (case sensitive). For example, `table_dict('exampleTableDict@Shared', 'office', SourceUserName)`.

"dict" function

Gets information about the value in the specified column of a dictionary of the dictionary type.

You must specify the parameters in the following sequence:

1. Dictionary name

2. One or more expressions whose results are used to generate the dictionary row key.

Usage example	Result
`dict('exampleDictionary', SourceAddress)`	Gets data from `exampleDictionary` from the row with the `SourceAddress` key.
`dict('exampleDictionary', SourceAddress, to_lower(SourceUserName))`	Gets data from the `exampleDictionary` from a composite key string from the `SourceAddress` field value and the lowercase value of the `SourceUserName` field.

Use this function to access the dictionary of the shared tenant from a variable. To do so, add the `@Shared` suffix after the name of the active list (case sensitive). For example, `dict('exampleDictionary@Shared', SourceAddress)`.

Operations with context tables

"context_table" function

Returns the value of the specified field in the base type (for example, integer, array of integers).

You must specify the parameters in the following sequence:

1. Name of the context table. The name must be specified.

2. Expression that returns the field name of context table.

3. Expression that returns the name of key field 1 of the context table.

4. Expression that returns the value of key field 1 of the context table.

The function must contain at least 4 parameters.

Usage example	Result
`context_table('tbl1', 'list_field1', 'key1', 'key1_val')`	`Get the value of the specified field. If the context table or context table field does not exist, an empty string is returned.`

"len" function

Returns the length of a string or array.

The function returns the length of the array if the passed array is of one of the following types:

- array of integers

- array of floats

- array of strings

- array of booleans

If an array of a different type is passed, the data of the array is cast to the string type, and the function returns the length of the resulting string.

Usage examples
`len(context_table('tbl1', 'list_field1', 'key1', 'key1_val'))`
`len(DeviceCustomString1)`

"distinct_items" function

Returns a list of unique elements in an array.

The function returns the list of unique elements of the array if the passed array is of one of the following types:

- array of integers
- array of floats
- array of strings
- array of booleans

If an array of a different type is passed, the data of the array is cast to the string type, and the function returns a string consisting of the unique characters from the original string.

Usage examples
`distinct_items(context_table('tbl1', 'list_field1', 'key1', 'key1_val'))`
`distinct_items(DeviceCustomString1)`

"sort_items" function

Returns a sorted list of array elements.

You must specify the parameters in the following sequence:

1. Expression that returns the object of the sorting
2. Sorting order Possible values: `asc`, `desc`. If the parameter is not specified, the default value is `asc`.

The function returns the list of sorted elements of the array if the passed array is of one of the following types:

- array of integers
- array of floats
- array of strings

For a boolean array, the function returns the list of array elements in the original order.

If an array of a different type is passed, the data of the array is cast to the string type, and the function returns a string of sorted characters.

Usage examples

`sort_items(context_table('tbl1', 'list_field1', 'key1', 'key1_val'), 'asc')`
`sort_items(DeviceCustomString1)`

"item" function

Returns the array element with the specified index or the character of a string with the specified index if an array of integers, floats, strings, or boolean values is passed.

You must specify the parameters in the following sequence:

1. Expression that returns the object of the indexing
2. Expression that returns the index of the element or character.

The function must contain at least 2 parameters.

The function returns the array element with the specified index or the string character with the specified index if the index falls within the range of the array and the passed array is of one of the following types:

- array of integers
- array of floats
- array of strings
- array of booleans

If an array of a different type is passed and the index falls within the range of the array, the data is cast to the string type, and the function returns the string character with the specified index. If an array of a different type is passed and the index is outside the range of the array, the function returns an empty string.

Usage examples
`item(context_table('tbl1', 'list_field1', 'key1', 'key1_val'), 1)`
`item(DeviceCustomString1, 0)`

Operation with rows

"len" function

Returns the number of characters in a string. 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
`len('SomeText')`
`len(Message)`
`len($otherVariable)`

"to_lower" function

Converts characters in a string to lowercase. Supported for standard fields and extended event schema fields of the "string" type.

A string can be passed as a string, field name or variable.

Usage examples
`to_lower(SourceUserName)`
`to_lower('SomeText')`
`to_lower($otherVariable)`

"to_upper" function

Converts characters in a string to uppercase. Supported for standard fields and extended event schema fields of the "string" type. A string can be passed as a string, field name or variable.

Usage examples
`to_upper(SourceUserName)`
`to_upper('SomeText')`
`to_upper($otherVariable)`

"append" function

Adds characters to the end of a string. Supported for standard fields and extended event schema fields of the "string" type.

You must specify the parameters in the following sequence:

1. Original string.
2. Added string.

Strings can be passed as a string, field name or variable.

Usage examples	Usage result
`append(Message, '123')`	The string `123` is added to the end of this string from the `Message` field.
`append($otherVariable, 'text')`	The string `text` is added to the end of this string from the variable `otherVariable`.
`append(Message, $otherVariable)`	A string from `otherVariable` is added to the end of this string from the `Message` field.

"prepend" function

Adds characters to the beginning of a string. Supported for standard fields and extended event schema fields of the "string" type.

You must specify the parameters in the following sequence:

1. Original string.
2. Added string.

Strings can be passed as a string, field name or variable.

Usage examples	Usage result
`prepend(Message, '123')`	The string `123` is added to the beginning of this string from the `Message` field.
`prepend($otherVariable, 'text')`	The string `text` is added to the beginning of this string from `otherVariable`.
`prepend(Message, $otherVariable)`	A string from `otherVariable` is added to the beginning of this string from the `Message` field.

"substring" function

Returns a substring from a string. Supported for standard fields and extended event schema fields of the "string" type.

You must specify the parameters in the following sequence:

1. Original string.
2. Substring start position (natural number or 0).
3. (Optional) substring end position.

Strings can be passed as a string, field name or variable. If the position number is greater than the original data string length, an empty string is returned.

Usage examples	Usage result
`substring(Message, 2)`	Returns a part of the string from the `Message` field: from 3 characters to the end.
`substring($otherVariable, 2, 5)`	Returns a part of the string from the `otherVariable` variable: from 3 to 6 characters.
`substring(Message, 0, len(Message) - 1)`	Returns the entire string from the `Message` field except the last character.

"tr" function

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. (Optional) string that should be removed from the beginning and end of the original string.

Strings can be passed as a string, field name or variable. If you do not specify a string to be deleted, spaces will be removed from the beginning and end of the original string.

Usage examples	Usage result
`tr(Message)`	Spaces have been removed from the beginning and end of the string from the `Message` field.
`tr($otherVariable, '_')`	If the `otherVariable` variable has the `_test_` value, the string `_test_` is returned.
`tr(Message, '@example.com')`	If the `Message` event field contains the string `user@example.com`, the string `user` is returned.

"replace" function

Replaces all occurrences of character sequence A in a string with character sequence B. Supported for standard fields and extended event schema fields of the "string" type.

You must specify the parameters in the following sequence:

1. Original string.

2. Search string: sequence of characters to be replaced.

3. Replacement string: sequence of characters to replace the search string.

Strings can be passed as an expression.

Usage examples	Usage result
`replace(Name, 'UserA', 'UserB')`	Returns a string from the `Name` event field in which all occurrences of `UserA` are replaced with `UserB`.
`replace($otherVariable, ' text ', '_text_')`	Returns a string from `otherVariable` in which all occurrences of `' text'` are replaced with `'_text_'`.

"regexp_replace" function

Replaces a sequence of characters that match a regular expression with a sequence of characters and regular expression capturing groups. Supported for standard fields and extended event schema fields of the "string" type.

You must specify the parameters in the following sequence:

1. Original string.

2. Search string: regular expression.

3. Replacement string: sequence of characters to replace the search string, and IDs of the regular expression capturing groups. A string can be passed as an expression.

Strings can be passed as a string, field name or variable. Unnamed capturing groups can be used.

In regular expressions used in variable functions, each backslash character must be additionally escaped. For example, `^example\\\\` must be used instead of the regular expression `^example\\`.

Usage examples	Usage result
`regexp_replace(SourceAddress, '([0-9]{1,3}).([0-9]{1,3}).([0-9]{1,3}).([0-9]{1,3})', 'newIP: $1.$2.$3.10')`	Returns a string from the `SourceAddress` event field in which the text `newIP` is inserted before the IP addresses. In addition, the last digits of the address are replaced with `10`.

"regexp_capture" function

Gets the result matching the regular expression condition from the original string. Supported for standard fields and extended event schema fields of the "string" type.

You must specify the parameters in the following sequence:

1. Original string.
2. Search string: regular expression.

Strings can be passed as a string, field name or variable. Unnamed capturing groups can be used.

In regular expressions used in variable functions, each backslash character must be additionally escaped. For example, `^example\\\\` must be used instead of the regular expression `^example\\`.

Usage examples	
`regexp_capture(Message, '(\\\\d{1,3}\\\\.\\\\d{1,3}\\\\.\\\\d{1,3}\\\\.\\\\d{1,3})')`	M ' f 1 s M ' f 4 t a 1 s

Operations with timestamps

now function

Gets a timestamp in epoch format. Runs with no arguments.

Usage examples
`now()`

"extract_from_timestamp" function

Gets atomic time representations (year, month, day, hour, minute, second, day of the week) from fields and variables with time in the epoch format.

The parameters must be specified in the following sequence:

1. Event field of the timestamp type, or variable.
2. Notation of the atomic time representation. This parameter is case sensitive.

   Possible variants of atomic time notation:

   - y refers to the year in number format.
   - M refers to the month in number notation.
   - d refers to the number of the month.
   - wd refers to the day of the week: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday.
   - h refers to the hour in 24-hour format.
   - m refers to the minutes.
   - s refers to the seconds.
3. (optional) Time zone notation. 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" function

Converts the time from RFC3339 format (for example, "2022-05-24 00:00:00", "2022-05-24 00:00:00+0300) to epoch format.

Usage examples
`parse_timestamp(Message)`
`parse_timestamp($otherVariable)`

"format_timestamp" function

Converts the time from epoch format to RFC3339 format.

The parameters must be specified in the following sequence:

1. Event field of the timestamp type, or variable.
2. Time format notation: RFC3339.
3. (optional) Time zone notation. If this parameter is not specified, the time is calculated in UTC format.

Usage examples
`format_timestamp(Timestamp, 'RFC3339')`
`format_timestamp($otherVariable, 'RFC3339')`
`format_timestamp(Timestamp, 'RFC3339', 'Europe/Moscow')`

"truncate_timestamp" function

Rounds the time in epoch format. After rounding, the time is returned in epoch format. Time is rounded down.

The parameters must be specified in the following sequence:

1. Event field of the timestamp type, or variable.
2. Rounding parameter:
   - 1s rounds to the nearest second.
   - 1m rounds to the nearest minute.
   - 1h rounds to the nearest hour.
   - 24h rounds to the nearest day.
3. (optional) Time zone notation. If this parameter is not specified, the time is calculated in UTC format.

Usage examples	Examples of rounded values	Usage result
`truncate_timestamp(Timestamp, '1m')`	1654631774175 (7 June 2022, 19:56:14.175)	1654631760000 (7 June 2022, 19:56:00)
`truncate_timestamp($otherVariable, '1h')`	1654631774175 (7 June 2022, 19:56:14.175)	1654628400000 (7 June 2022, 19:00:00)
`truncate_timestamp(Timestamp, '24h', 'Europe/Moscow')`	1654631774175 (7 June 2022, 19:56:14.175)	1654560000000 (7 June 2022, 0:00:00)

"time_diff" function

Gets the time interval between two timestamps in epoch format.

The parameters must be specified in the following sequence:

1. Interval end time. Event field of the timestamp type, or variable.
2. Interval start time. Event field of the timestamp type, or variable.
3. Time interval notation:
   - ms refers to milliseconds.
   - s refers to seconds.
   - m refers to minutes.
   - h refers to hours.
   - d refers to days.

Usage examples
`time_diff(EndTime, StartTime, 's')`
`time_diff($otherVariable, Timestamp, 'h')`
`time_diff(Timestamp, DeviceReceiptTime, 'd')`

Mathematical operations

These are comprised of basic mathematical operations and functions.

Basic mathematical operations

Supported for integer and float fields of the extended event schema.

Operations:

- Addition
- Subtraction
- Multiplication
- Division
- Modulo division

Parentheses determine the sequence of actions

Available arguments:

- Numeric event fields
- Numeric variables
- Real numbers

  When modulo dividing, only natural numbers can be used as arguments.

Usage constraints:

- Division by zero returns zero.
- Mathematical operations between numbers and strings return zero.
- Integers resulting from operations are returned without a dot.

Usage examples **`(Type=3; otherVariable=2; Message=text)`**	Usage result
`Type + 1`	4
`$otherVariable - Type`	-1
`2 * 2.5`	5
`2 / 0`	0
`Type * Message`	0
`(Type + 2) * 2`	10
`Type % $otherVariable`	1

"round" function

Rounds numbers. Supported for integer and float fields of the extended event schema.

Available arguments:

- Numeric event fields
- Numeric variables
- Numeric constants

Usage examples **`(DeviceCustomFloatingPoint1=7.75; DeviceCustomFloatingPoint2=7.5 otherVariable=7.2)`**	Usage result
`round(DeviceCustomFloatingPoint1)`	8
`round(DeviceCustomFloatingPoint2)`	8
`round($otherVariable)`	7

"ceil" function

Rounds up numbers. Supported for integer and float fields of the extended event schema.

Available arguments:

- Numeric event fields
- Numeric variables
- Numeric constants

Usage examples **`(DeviceCustomFloatingPoint1=7.15; otherVariable=8.2)`**	Usage result
`ceil(DeviceCustomFloatingPoint1)`	8
`ceil($otherVariable)`	9

"floor" function

Rounds down numbers. Supported for integer and float fields of the extended event schema.

Available arguments:

- Numeric event fields
- Numeric variables
- Numeric constants

Usage examples **`(DeviceCustomFloatingPoint1=7.15; otherVariable=8.2)`**	Usage result
`floor(DeviceCustomFloatingPoint1)`	7
`floor($otherVariable)`	8

"abs" function

Gets the modulus of a number. Supported for integer and float fields of the extended event schema.

Available arguments:

- Numeric event fields
- Numeric variables
- Numeric constants

Usage examples **`(DeviceCustomNumber1=-7; otherVariable=-2)`**	Usage result
`abs(DeviceCustomFloatingPoint1)`	7
`abs($otherVariable)`	2

"pow" function

Exponentiates a number. Supported for integer and float fields of the extended event schema.

The parameters must be specified in the following sequence:

1. Base — real numbers.
2. Power — natural numbers.

Available arguments:

- Numeric event fields
- Numeric variables
- Numeric constants

Usage examples
`pow(DeviceCustomNumber1, DeviceCustomNumber2)`
`pow($otherVariable, DeviceCustomNumber1)`

"str_join" function

Join multiple strings into one using a separator. Supported for integer and float fields of the extended event schema.

The parameters must be specified in the following sequence:

1. Separator. String.
2. String1, string2, stringN. At least 2 expressions.

Usage examples	Usage result
`str_join('|', to_lower(Name), to_upper(Name), Name)`	String.

"conditional" function

Get one value if a condition is met and another value if the condition is not met. Supported for integer and float fields of the extended event schema.

The parameters must be specified in the following sequence:

1. Condition. String. The syntax is similar to the conditions of the Where statement in SQL. You can use the functions of the KUMA variables and references to other variables in a condition.
2. The value if the condition is met. Expression.
3. The value if the condition is not met. Expression.

Supported operators:

- AND
- OR
- NOT
- =
- !=
- <

- <=
- >
- >=
- LIKE (RE2 regular expression is used, rather than an SQL expression)
- ILIKE (RE2 regular expression is used, rather than an SQL expression)
- BETWEEN
- IN
- IS NULL (check for an empty value, such as 0 or an empty string)

Usage examples (the value depends on arguments 2 and 3)
conditional('SourceUserName = \\'root\\' AND DestinationUserName = SourceUserName', 'match', 'no match')
conditional(`DestinationUserName ILIKE 'svc_.*'`, 'match', 'no match')
conditional(`DestinationUserName NOT LIKE 'svc_.*'`, 'match', 'no match')

Operations for extended event schema fields

For extended event schema fields of the "string" type, the following kinds of operations are supported:

- "len" function
- "to_lower" function
- "to_upper" function
- "append" function
- "prepend" function
- "substring" function
- "tr" function
- "replace" function
- "regexp_replace" function
- "regexp_capture" function

For extended event schema fields of the integer or float type, the following kinds of mathematical operations are supported:

- Basic mathematical operations:
- "round" function

- "ceil" function
- "floor" function
- "abs" function
- "pow" function
- "str_join" function
- "conditional" function

For extended event schema fields of the "array of numbers", "array of floats", and "array of strings" types, the following kinds of mathematical operations are supported:

- item(SA.someStringArray, i) — gets the i-th element of the someStringArray[i] field.
- SA.someStringArray , returns ["string1", "string2", "string1"] — gets the array of values from the someStringArray field.
- len(SA.someStringArray) — gets the number of elements in the someStringArray array.
- distinct_items(SA.someStringArray), returns ["string1", "string2"] — gets unique elements from the someStringArray array.
- to_string(SA.someStringArray) — generates a TSV string from the array.
- sort_items_<type>(SA.someStringArray, ASC); instead of <type>, you must specify the array type: 'sa' for an array of strings, 'fa' for an array of floats, 'na' for an array of integers. Example: sort_item_sa.

For fields of the "array of integers" and "array of floats" types, the following functions are supported:

• min_na — returns the minimum element of an array of numbers.

• max_na — returns the maximum element of an array of numbers.

• avg_na — returns the average value of the array.

• min_fa — returns the minimum element of an array of floats.

• max_fa — returns the maximum element of an array of floats.

• avg_fa — returns the average value of an array of floats.

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.

     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.

     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.

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] Granted TGS without TGT (Golden Ticket)	Detects suspected "Golden Ticket" type attacks. This rule works based on Microsoft Windows events.
[OOTB][AD][Technical] 4768. TGT Requested	The technical rule used to populate the active list is [OOTB][AD] List of requested TGT. EventID 4768. This rule works based on Microsoft Windows events.
[OOTB][AD] Membership of sensitive group was modified	Works based on Microsoft Windows events.
[OOTB][AD] Multiple accounts failed to log on from the same host	Triggers after multiple failed authentication attempts are detected on the same host from different accounts.
[OOTB][AD] Possible Kerberoasting attack	Detects suspected "Kerberoasting" type attacks. This rule works based on Microsoft Windows events.
[OOTB][AD] Successful authentication with the same account on multiple hosts	Detects connections to different hosts under the same account. This rule works based on Microsoft Windows events.
[OOTB][AD] The account added and deleted from the group in a short period 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.

Filters

Filters let you select events based on specified conditions.

The collector service uses filters to filter out events that you do not want to send to KUMA. That is, an event that matches the filter condition is NOT sent to KUMA.

Filters can be used in the following KUMA services and features:

- Collector.
- Correlator.
- Storage.
- KUMA agents.
- Correlation rules.
- Enrichment rules.
- Aggregation rules.
- Destinations.
- Response rules.
- Segmentation rules.

You can use standalone filters or built-in filters that are stored in the service or resource where they were created.

For these resources, you can enable the display of control characters in all input fields except the **Description** field.

Available settings for filters:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters. Inline filters are created in other resources or services and do not have names.
- **Tenant** (required)—name of the tenant that owns the resource.
- Description—up to 4,000 Unicode characters describing the filter.
- The **Conditions** group of settings lets you formulate filtering criteria by creating filter conditions and groups of filters, or by adding existing filters.

  To create filtering criteria, you can use *builder mode* or *source code mode*. The builder mode is used by default.

  In builder mode, you can create or edit filter criteria by selecting filter conditions and operators from drop-down lists.

  In source code mode, you can use text commands to create and edit search queries.

You can freely switch between modes when creating filtering criteria. To switch to source code mode, click the **Code** button. When switching between modes, the created condition filters are preserved. If the filter code is not displayed on the **Code** tab after linking the created filter to the resource, go to the **Builder** tab and then go back to the **Code** tab. The filter code is displayed.

Creating conditions in builder mode

You can create filtering criteria in builder mode using the following buttons:

- **Add condition** adds a string with fields for defining a condition.
- **Add group** adds a group of filters. Group operators can be switched between **AND**, **OR**, and **NOT**. You can add groups, conditions, and existing filters to groups of filters. Conditions placed in the **NOT** subgroup are combined with the **AND** operator.

To replace an operator in the created condition, click the operator that you want to replace and select the new operator from the drop-down list.

To delete an operator in the created condition, click the operator that you want to delete and press `Backspace`.

To alter the sequence of filter conditions, click ⠿ and drag and drop the condition to the new location.

Conditions, groups, and filters can be deleted by using the × button.

Settings of conditions:

- **When** (required)—in this drop-down list, you can specify whether or not to use the inverted function of the operator.
- **Left operand** and **Right operand** (required)—used to specify the values that the operator will process. The available types depend on the selected operator.

  Operands of filters

- **Event field**—used to assign an event field value to the operand. Advanced settings:
  - **Event field** (required)—this drop-down list is used to select the field from which the value for the operand should be extracted.
- **Active list**—used to assign an active list record value to the operand. Advanced settings:
  - **Active list** (required)—this drop-down list is used to select the active list.
  - **Key fields** (required)—this is the list of event fields used to create the Active list entry and serve as the Active list entry key.
  - **Field** (required unless the **inActiveList** operator is selected)—used to enter the Active list field name from which the value for the operand should be extracted.
- **Context table**—used to assign a context table value to the operand. Advanced settings:
  - **Context table name** (required)—this drop-down list is used to select a context table.
  - **key fields** (required)—a list of event fields or local variable that are used to create a context table record and serve as the key for the context table record.
  - **field**—used to enter the name of the context table field from which the operand value must be extracted.
  - **index**—used to enter the index of the list field of the table from which the operand value must be extracted.
- **Dictionary**—used to assign a dictionary resource value to the operand. Advanced settings:
  - **Dictionary** (required)—this drop-down list is used to select the dictionary.
  - **Key fields** (required)—this is the list of the event fields used to form the dictionary value key.
- **Constant**—used to assign a custom value to the operand. Advanced settings:
  - **Value** (required)—here you enter the constant that you want to assign to the operand.
- **Table**—used to assign multiple custom values to the operand. Advanced settings:
  - **Dictionary** (required)—this drop-down list is used to select a **Table**-type dictionary.
  - **Key fields** (required)—this is the list of the event fields used to form the dictionary value key.
- **List**—used to assign multiple custom values to the operand. Advanced settings:
  - **Value** (required)—here you enter the list of 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**—used to read the CyberTrace threat intelligence (TI) data from the events. Advanced settings:
  - **Feed** (required)—this field is used to specify the CyberTrace threat category.

- **Key fields** (required)—this drop-down list is used to select the event field containing the CyberTrace threat indicators.
- **Field** (required)—this field is used to specify the CyberTrace feed field containing the threat indicators.

- **Operator** (required)—used to select the condition operator.

  In this drop-down list, you can select the **do not match case** check box if the operator should ignore the case of values. This check box is ignored if the **inSubnet**, **inActiveList**, **inCategory**, **InActiveDirectoryGroup, hasBit**, **inDictionary** operators are selected. This check box is cleared by default.

  Filter operators ?

- **=**—the left operand equals the right operand.

- **<**—the left operand is less than the right operand.

- **<=**—the left operand is less than or equal to the right operand.

- **>**—the left operand is greater than the right operand.

- **>=**—the left operand is greater than or equal to the right operand.

- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).

- **contains**—the left operand contains values of the right operand.

- **startsWith**—the left operand starts with one of the values of the right operand.

- **endsWith**—the left operand ends with one of the values of the right operand.

- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.

- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.

- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.

- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.

- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.

- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

The available operand kinds depends on whether the operand is left (L) or right (R).

Available operand kinds for left (L) and right (R) operands

Operator	Event field type	Active list type	Dictionary type	Context table type	Table type	TI type	Constant type	List type
=	L,R	L,R	L,R	L,R	L,R	L,R	R	R
>	L,R	L,R	L,R	L,R (only when looking up a table value by index)	L,R	L	R	–
>=	L,R	L,R	L,R	L,R (only when looking up a table value by index)	L,R	L	R	–
<	L,R	L,R	L,R	L,R (only when looking up a table value by index)	L,R	L	R	–
<=	L,R	L,R	L,R	L,R (only when looking up a table value by index)	L,R	L	R	–
inSubnet	L,R	L,R	L,R	L,R	L,R	L,R	R	R
contains	L,R	L,R	L,R	L,R	L,R	L,R	R	R
startsWith	L,R	L,R	L,R	L,R	L,R	L,R	R	R
endsWith	L,R	L,R	L,R	L,R	L,R	L,R	R	R
match	L	L	L	L	L	L	R	R
hasVulnerability	L	L	L	L	L	–	–	–
hasBit	L	L	L	L	L	–	R	R
inActiveList	–	–	–	–	–	–	–	–
inDictionary	–	–	–	–	–	–	–	–
inCategory	L	L	L	L	L	–	R	R
inContextTable	–	–	–	–	–	–	–	–
inActiveDirectoryGroup	L	L	L	L	L	–	R	R
TIDetect	–	–	–	–	–	–	–	–

You can use hotkeys when managing filters. Hotkeys are described in the table below.

Hotkeys and their functions

Key	Function
e	Invokes a filter by the event field
d	Invokes a filter by the dictionary field
a	Invokes a filter by the active list field
c	Invokes a filter by the context table field

t	Invokes a filter by the table field
f	Invokes a filter
t+i	Invokes a filter using TI
Ctrl+Enter	Finish editing a condition

The usage of extended event schema fields "string", "number", or "float" types is the same as the usage of fields of the KUMA event schema.

When using filters with extended event schema fields of the "Array of strings", "Array of numbers", and "Array of floats" types, you can use the following operations:

- The "contains" operation returns True if the specified substring is present in the array, otherwise it returns False.
- The "match" operation matches the string against a regular expression.
- The "intersec" operation.

When using filters with extended event schema fields of the "Array of numbers" and "Array of floats" types, you can use the following comparison operations: <, >, =, >=, <=.

If you want to reference a specific element of an array in the filter, you can use the following syntax: NA.<array name>.<index of the element>

Arrays are 0-based.

Example:

NA.ArrayOne.0 — access to the first element of the ArrayOne array of integers.

FA.ArrayTwo.2 — access to the third element of the ArrayTwo array of floats.

Creating conditions in source code mode

The code editor mode allows you to quickly edit conditions, select and copy blocks of code.

On the right side of the builder, you can find the navigator, which lets you to navigate the filter code.

Line wrapping is performed automatically at AND, OR, NOT logical operators, or at commas that delimit the items in the list of values.

Names of resources used in the filter are automatically specified. Fields containing the names of linked resources cannot be edited. The names of shared resource categories are not displayed in the filter if you do not have the "Access to shared resources" role.

The filters listed in the table below are included in the 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.

Only data based on correlation rules of the correlator are added to the active list.

You can add, edit, duplicate, delete, and export records in the active correlator sheet.

During the correlation process, when entries are deleted from active lists, service events are generated in the correlators. These events only exist in the correlators, and they are not redirected to other destinations. Correlation rules can be configured to track these events so that they can be used to identify threats. Service event fields for deleting an entry from the active list are described below.

Event field	Value or comment
`ID`	Event identifier
`Timestamp`	Time when the expired entry was deleted
`Name`	`"active list record expired"`
`DeviceVendor`	`"Kaspersky"`
`DeviceProduct`	`"KUMA"`
`ServiceID`	Correlator ID
`ServiceName`	Correlator name
`DeviceExternalID`	Active list ID
`DevicePayloadID`	Key of the expired entry
`BaseEventCount`	Number of deleted entry updates increased by one

Viewing the table of active lists

To view the table of correlator active lists:

1. In the KUMA web interface, select the **Resources** section.
2. In the **Services** section, click the **Active services** button.
3. Select the check box next to the correlator for which you want to view the active list.
4. Click the **Go to active lists** button.

   The **Correlator active lists** table is displayed.

The table contains the following data:

- **Name**—the name of the correlator list.
- **Records**—the number of record the active list contains.
- **Size on disk**—the size of the active list.
- **Directory**—the path to the active list on the KUMA Core server.

Adding active list

To add active list:

1. In the KUMA web interface, select the **Resources** section.
2. In the **Resources** section, click the **Active lists** button.
3. Click the **Add active list** button.
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 the record added to the active list is stored in it.
      When the specified time expires, the record is deleted. The time is specified in seconds.
      The default value is 0. If the value of the field is 0, the record is retained for 36,000 days (roughly 100 years).
   d. In the **Description** field, provide any additional information.
      You can use up to 4,000 Unicode characters.
      This field is optional.
5. Click the **Save** button.

The active list is added.

Viewing the settings of an active list

To view the settings of an active list:

1. In the KUMA web interface, select the **Resources** section.
2. In the **Resources** section, click the **Active lists** button.
3. In the **Name** column, select the active list whose settings you want to view.

This opens the active list settings window. It displays the following information:

- **ID**—identifier selected Active list.
- **Name**—unique name of the resource.
- **Tenant**—the name of the tenant that owns the resource.
- **TTL**—the record added to the active list is stored in it for this time. This value is specified in seconds.
- **Description**—any additional information about the resource.

Changing the settings of an active list

To change the settings of an active list:

1. In the KUMA web interface, select the **Resources** section.
2. In the **Resources** section, click the **Active lists** button.
3. In the **Name** column, select the active list whose settings you want to change.
4. Specify the values of the following parameters:
   - **Name**—unique name of the resource.
   - **TTL**—the record added to the active list is stored in it for this time. This value is specified in seconds.
     If the field is set to 0, the record is stored indefinitely.
   - **Description**—any additional information about the resource.

   > The **ID** and **Tenant** fields are not editable.

Duplicating the settings of an active list

To copy an active list:

1. In the KUMA web interface, select the **Resources** section.
2. In the **Resources** section, click the **Active lists** button.
3. Select the check box next to the active lists you want to copy.
4. Click **Duplicate**.
5. Specify the necessary settings.
6. Click the **Save** button.

The active list is copied.

Deleting an active list

To delete an active list:

1. In the KUMA web interface, select the **Resources** section.
2. In the **Resources** section, click the **Active lists** button.
3. Select the check boxes next to the active lists you want to delete.
   To delete all lists, select the check box next to the **Name** column.

> At least one check box must be selected.

4. Click the **Delete** button.
5. Click **OK**.

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 **Resources** section.
2. In the **Services** section, click the **Active services** button.
3. Select the check box next to the correlator for which you want to view the active list.
4. Click the **Go to active lists** button.

   The **Correlator active lists** table is displayed.
5. In the **Name** column, select the desired active list.

A table of records for the selected list is opened.

The table contains the following data:

- **Key** – the value of the record key.
- **Record repetitions** – total number of times the record was mentioned in events and identical records were downloaded when importing active lists to KUMA.
- **Expiration date** – date and time when the record must be deleted.

  If the **TTL** field had the value of 0 when the active list was created, the records of this active list are retained for 36,000 days (roughly 100 years).
- **Created** – the time when the active list was created.
- **Updated** – the time when the active list was last updated.

Searching for records in the active list

To find a record in the active list:

1. In the KUMA web interface, select the **Resources** section.
2. In the **Services** section, click the **Active services** button.
3. Select the check box next to the correlator for which you want to view the active list.

4. Click the **Go to active lists** button.

   The **Correlator active lists** table is displayed.

5. In the **Name** column, select the desired active list.

   A window with the records for the selected list is opened.

6. In the **Search** field, enter the record key value or several characters from the key.

The table of records of the active list displays only the records with the key containing the entered characters.

Adding a record to an active list

To add a record to the active list:

1. In the KUMA web interface, select the **Resources** section.

2. In the **Services** section, click the **Active services** button.

3. Select the check box next to the required correlator.

4. Click the **Go to active lists** button.

   The **Correlator active lists** table is displayed.

5. In the **Name** column, select the desired active list.

   A window with the records for the selected list is opened.

6. Click **Add**.

   The **Create record** window opens.

7. Specify the values of the following parameters:

   a. In the **Key** field, enter the name of the record.

      You can specify several values separated by the "|" character.

      > The **Key** field cannot be empty. If the field is not filled in, KUMA returns an error when trying to save the changes.

   b. In the **Value** field, specify the values for fields in the **Field** column.

      KUMA takes field names from the correlation rules with which the active list is associated. These names are not editable. You can delete these fields if necessary.

   c. Click the **Add new element** button to add more values.

   d. In the **Field** column, specify the field name.

      The name must meet the following requirements:

      - To be unique
      - Do not contain tab characters

- Do not contain special characters except for the underscore character
- The maximum number of characters is 128.

> The name must not begin with an underscore and contain only numbers.

e. In the **Value** column, specify the value for this field.
It must meet the following requirements:

- Do not contain tab characters
- Do not contain special characters except for the underscore character
- The maximum number of characters is 1024.

This field is optional.

8. Click the **Save** button.

The record is added. After saving, the records in the active list are sorted in alphabet order.

Duplicating records in the active list

To duplicate a record in the active list:

1. In the KUMA web interface, select the **Resources** section.
2. In the **Services** section, click the **Active services** button.
3. Select the check box next to the correlator for which you want to view the active list.
4. Click the **Go to active lists** button.
   The **Correlator active lists** table is displayed.
5. In the **Name** column, select the desired active list.
   A window with the records for the selected list is opened.
6. Select the check boxes next to the record you want to copy.
7. Click **Duplicate**.
8. Specify the necessary settings.

> The **Key** field cannot be empty. If the field is not filled in, KUMA returns an error when trying to save the changes.

> Editing the field names in the **Field** column is not available for the records that have been added to the active list before. You can change the names only for records added at the time of editing. The name must not begin with an underscore and contain only numbers.

9. Click the **Save** button.

The record is copied. After saving, the records in the active list are sorted in alphabet order.

Changing a record in the active list

To edit a record in the active list:

1. In the KUMA web interface, select the **Resources** section.
2. In the **Services** section, click the **Active services** button.
3. Select the check box next to the correlator for which you want to view the active list.
4. Click the **Go to active lists** button.

   The **Correlator active lists** table is displayed.
5. In the **Name** column, select the desired active list.

   A window with the records for the selected list is opened.
6. Click the record name in the **Key** column.
7. Specify the required values.
8. Click the **Save** button.

The record is overwritten. After saving, the records in the active list are sorted in alphabet order.

Restrictions when editing a record:

- The record name is not editable. You can change it by importing the same data with a different name.
- Editing the field names in the **Field** column is not available for the records that have been added to the active list before. You can change the names only for records added at the time of editing. The name must not begin with an underscore and contain only numbers.
- The values in the **Value** column must meet the following requirements:
  - Do not contain Cyrillic characters
  - Do not contain spaces or tabs
  - Do not contain special characters except for the underscore character
  - The maximum number of characters is 128

Deleting records from the active list

To delete records from the active list:

1. In the KUMA web interface, select the **Resources** section.

2. In the **Services** section, click the **Active services** button.

3. Select the check box next to the correlator for which you want to view the active list.

4. Click the **Go to active lists** button.

   The **Correlator active lists** table is displayed.

5. In the **Name** column, select the desired active list.

   A window with the records for the selected list is opened.

6. Select the check boxes next to the records you want to delete.

   To delete all records, select the check box next to the **Key** column.

   > At least one check box must be selected.

7. Click the **Delete** button.

8. Click **OK**.

The records will be deleted.

Import data to an active list

To import active list:

1. In the KUMA web interface, select the **Resources** section.

2. In the **Services** section, click the **Active services** button.

3. Select the check box next to the correlator for which you want to view the active list.

4. Click the **Go to active lists** button.

   The **Correlator active lists** table is displayed.

5. Point the mouse over the row with the desired active list.

6. Click ... to the left of the active list name.

7. Select **Import**.

   The active list import window opens.

8. In the **File** field select the file you wan to import.

9. In the **Format** drop-down list select the format of the file:

   - **csv**
   - **tsv**
   - **internal**

10. Under **Key field**, enter the name of the column containing the active list record keys.

11. Click the **Import** button.

The data from the file is imported into the active list. The records included in the list before are saved.

Data imported from a file is not checked for invalid characters. If you use this data in widgets, widgets are displayed incorrectly if invalid characters are present in the data.

Exporting data from the active list

To export active list:

1. In the KUMA web interface, select the **Resources** section.

2. In the **Services** section, click the **Active services** button.

3. Select the check box next to the correlator for which you want to view the active list.

4. Click the **Go to active lists** button.

   The **Correlator active lists** table is displayed.

5. Point the mouse over the row with the desired active list.

6. Click ... to the left of the desired active list.

7. Click the **Export** button.

The active list is downloaded in the JSON format using your browsers settings. The name of the downloaded file reflects the name of active list.

Predefined active lists

The active lists listed in the table below are included in the 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 requested TGT. EventID 4768	This active list is populated by the "[OOTB][AD][Technical] 4768. TGT Requested" rule, this active list is also used in the selector of the "[OOTB][AD] Granted TGS without TGT (Golden Ticket)" rule. Records are removed from the list 10 hours after they are recorded.
[OOTB][AD] List of sensitive groups	This active list is used as a filter for the "[OOTB][AD] Membership of sensitive group was modified" correlation rule. Critical domain groups, whose membership must be monitored, can be added to the active list. Records are not deleted from the active list.

[OOTB][Linux] CompromisedHosts	This active list is populated by the [OOTB] Successful Bruteforce by potentially compromised Linux hosts rule. Records are removed from the list 24 hours after they are recorded.

Dictionaries

Description of parameters

Dictionaries are resources storing data that can be used by other KUMA resources and services.

Dictionaries can be used in the following KUMA services and features:

- Collector.
- Correlation rules.
- Normalizers.

Available settings:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Description**—up to 4,000 Unicode characters describing the resource.
- **Type** (required)—type of dictionary. The selected type determines the format of the data that the dictionary can contain:
  - You can add key-value pairs to the **Dictionary** type.

    It is not recommended to add more than 50,000 entries to dictionaries of this type.

    > When adding lines with the same keys to the dictionary, each new line will overwrite the existing line with the same key. This means that only one line will be added to the dictionary.

  - Data in the form of complex tables can be added to the **Table** type. You can interact with this type of dictionary by using the REST API.
- **Values** settings block—contains a table of dictionary data:

  For the **Dictionary** type, this block displays a list of **Key**—**Value** pairs. You can use the + button to add rows to the table. You can delete rows by using the × button that appears when you hover your mouse cursor over a row. In the **Key** field, you may optionally specify a unique value: up to 128 Unicode characters, the first character may not be $. In the **Value** field, you may optionally specify a value: up to 255 Unicode characters, the first character may not be $. You may add one or more **Key**–**Value** pairs.

  - For the **Table** type, this block displays a table containing data. You can use the + button to add rows and columns to the table. You can delete rows and columns by using the × buttons that are displayed when you hover your mouse cursor over a row or a column header. Column headers can be edited.

If the dictionary contains more than 5,000 entries, they are not displayed in the KUMA 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, Active Directory, and running a custom script for specific events.

Automatic execution of Kaspersky Security Center tasks, Kaspersky Endpoint Detection and Response tasks, and KICS for Networks and Active Directory tasks in accordance with response rules is available when integrated with the relevant programs.

You can configure response rules under **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.

Response rule settings

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
Type	Required setting, available if KUMA is integrated with Kaspersky Security Center. Response rule 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
Workers	The number of processes that the service can run simultaneously. By default, the number of workers is the same as the number of virtual processors on the server where the service is installed.
Description	Description of the response rule. You can add up to 4,000 Unicode characters.
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**

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. The name must contain 1 to 128 Unicode characters.
4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

   Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

   Filter operators ?

- **=**—the left operand equals the right operand.

- **<**—the left operand is less than the right operand.

- **<=**—the left operand is less than or equal to the right operand.

- **>**—the left operand is greater than the right operand.

- **>=**—the left operand is greater than or equal to the right operand.

- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).

- **contains**—the left operand contains values of the right operand.

- **startsWith**—the left operand starts with one of the values of the right operand.

- **endsWith**—the left operand ends with one of the values of the right operand.

- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.

- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.

- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.

- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.

- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.

- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have

completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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.

Response rules for a custom script

You can create a script containing commands to be executed on the KUMA server when selected events are detected and configure response rules to automatically run this script. In this case, the program will run the script when it receives events that match the response rules.

The script file is stored on the server where the correlator service using the response resource is installed: /opt/kaspersky/kuma/correlator/<Correlator ID>/scripts. The `kuma` user of this server requires the permissions to run the script.

When creating and editing response rules for a custom script, you need to define values for the following parameters.

Response rule settings

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
Type	Required setting. Response rule type, **script**.
Timeout	The number of seconds allotted for the script to finish. If this amount of time is exceeded, the script is terminated.
Script name	Required setting. Name of the script file. If the response resource is attached to the correlator service but there is no script file in the /opt/kaspersky/kuma/correlator/<Correlator ID>/scripts folder, the correlator will not work.
Script arguments	Arguments or event field values that must be passed to the script. If the script includes actions taken on files, you should specify the absolute path to these files. Parameters can be written with quotation marks ("). Event field names are passed in the `{{.EventField}}` format, where `EventField` is the name of the event field which value must be passed to the script. Example: `-n "\"usr\": {{.SourceUserName}}"`
Workers	The number of processes that the service can run simultaneously. By default, the number of workers is the same as the number of virtual processors on the server where the service is installed.
Description	Description of the resource. You can add up to 4,000 Unicode characters.
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** ?

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. The name must contain 1 to 128 Unicode characters.
4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

   Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

   Filter operators ?

- **=**—the left operand equals the right operand.

- **<**—the left operand is less than the right operand.

- **<=**—the left operand is less than or equal to the right operand.

- **>**—the left operand is greater than the right operand.

- **>=**—the left operand is greater than or equal to the right operand.

- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).

- **contains**—the left operand contains values of the right operand.

- **startsWith**—the left operand starts with one of the values of the right operand.

- **endsWith**—the left operand ends with one of the values of the right operand.

- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.

- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.

- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.

- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.

- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.

- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have

completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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.

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.

Response rule settings

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
Type	Required setting. Response rule type, **kics**.
Event field	Required setting. Specifies the event field for the asset for which response actions must be performed. Possible values: • SourceAssetID

	- DestinationAssetID - DeviceAssetID
KICS for Networks task	Response action to be performed when data is received that matches the filter. The following types of response actions are available: - **Change asset status to Authorized**. - **Change asset status to Unauthorized**. When a response rule is triggered, KUMA will send KICS for Networks an API request to change the status of the specified device to **Authorized** or **Unauthorized**.
Workers	The number of processes that the service can run simultaneously. By default, the number of workers is the same as the number of virtual processors on the server where the service is installed.
Description	Description of the resource. You can add up to 4,000 Unicode characters.
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

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. The name must contain 1 to 128 Unicode characters.
4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

   Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

   Filter operators ?

- **=**—the left operand equals the right operand.

- **<**—the left operand is less than the right operand.

- **<=**—the left operand is less than or equal to the right operand.

- **>**—the left operand is greater than the right operand.

- **>=**—the left operand is greater than or equal to the right operand.

- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).

- **contains**—the left operand contains values of the right operand.

- **startsWith**—the left operand starts with one of the values of the right operand.

- **endsWith**—the left operand ends with one of the values of the right operand.

- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.

- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.

- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.

- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.

- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.

- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have

	completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand. d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list. f. You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a search 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.

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.

Response rule settings

Setting	Description
Event field	Required setting. Specifies the event field for the asset for which response actions must be performed. Possible values: • SourceAssetID • DestinationAssetID • DeviceAssetID
Task type	Response action to be performed when data is received that matches the filter. The following types of response actions are available:

- Enable network isolation. When selecting this type of response, you need to define values for the following setting:
  - **Isolation timeout**—the number of hours during which the network isolation of an asset will be active. You can indicate from 1 to 9,999 hours. If necessary, you can add an exclusion for network isolation ?.

    To add an exclusion for network isolation:

    a. Click the **Add exclusion** button.

    b. Select the direction of network traffic that must not be blocked:

    - Inbound.
    - Outbound.
    - Inbound/Outbound.

    c. In the **Asset IP** field, enter the IP address of the asset whose network traffic must not be blocked.

    d. If you selected **Inbound** or **Outbound**, specify the connection ports in the **Remote ports** and **Local ports** fields.

    e. If you want to add more than one exclusion, click **Add exclusion** and repeat the steps to fill in the **Traffic direction**, **Asset IP**, **Remote ports** and **Local ports** fields.

    f. If you want to delete an exclusion, click the **Delete** button under the relevant exclusion.

  When adding exclusions to a network isolation rule, Kaspersky Endpoint Detection and Response may incorrectly display the port values in the rule details. This does not affect application performance. For more details on viewing a network isolation rule, please refer to the *Kaspersky Anti Targeted Attack Platform Help Guide*.

- Disable network isolation.

- Add prevention rule. When selecting this type of response, you need to define values for the following settings:
  - **Event fields to extract hash from**—event fields from which KUMA extracts SHA256 or MD5 hashes of files that must be prevented from running.
    The selected event fields, as well as the values selected in **Event field**, must be added to the propagated fields of the correlation rule.
  - **File hash #1**—SHA256 or MD5 hash of the file to be blocked.

  At least one of the above fields must be completed.

- Delete prevention rule.

- Run program. When selecting this type of response, you need to define values for the following settings:

	• **File path**—path to the file of the process that you want to start. • **Command line parameters**—parameters with which you want to start the file. • **Working directory**—directory in which the file is located at the time of startup. When a response rule is triggered for users with the General Administrator role, the **Run program** task will be displayed in the **Task manager** section of the program web interface. **Scheduled task** is displayed for this task in the **Created** column of the task table. You can view task completion results. All of the listed operations can be performed on assets that have Kaspersky Endpoint Agent for Windows. On assets that have Kaspersky Endpoint Agent for Linux, the program can only be started. At the software level, the capability to create prevention rules and network isolation rules for assets with Kaspersky Endpoint Agent for Linux is unlimited. KUMA and Kaspersky Endpoint Detection and Response do not provide any notifications about unsuccessful application of these rules.
Workers	The number of processes that the service can run simultaneously. By default, the number of workers is the same as the number of virtual processors on the server where the service is installed.
Description	Description of the response rule. You can add up to 4,000 Unicode characters.
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**

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. The name must contain 1 to 128 Unicode characters.
4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

   Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

   Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have

completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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.

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.

Response rule settings

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
Type	Required setting. Response rule type, **Response via Active Directory**.
Account ID source	Event field from which the Active Directory account ID value is taken. Possible values: • SourceAccountID • DestinationAccountID
AD	Command that is applied to the account when the response rule is triggered.

command	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 = HQ Team, OU = Groups, OU = ExchangeObjects, DC = avp, DC = ru. 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 = HQ Team, OU = Groups, OU = ExchangeObjects, DC = avp, DC = ru. 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
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

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. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters.

      Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters that you need to use to define the value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select the relevant operator.

      Filter operators ?

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns *False*.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In

other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If necessary, select the **do not match case** check box. When this check box is selected, the operator ignores the case of the values.

The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators.

This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

f. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a search 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.

Notification templates

Notification templates are used in alert generation notifications.

Notification template settings

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
Subject	Subject of the email containing the notification about the alert generation. In the email subject, you can refer to the alert fields. Example: `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.
Template	Required setting. The body of the email containing the notification about the alert generation. The template supports a syntax that can be used to populate the notification with data from the alert. You can read more about the syntax in the official Go language documentation.

	For convenience, you can open the email 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 **Save** to save the changes and close the window.

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.

Functions in notification templates

Functions available in templates are listed in the table below.

Functions in templates

Setting	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"`

`limit`	The 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: `{{ link_alert }}`
`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" }}`

Notification template syntax

In a template, you can query the alert fields containing a string or number:

```
{{ .CorrelationRuleName }}
```

The message will display the alert name, which is the contents of the `CorrelationRuleName` field.

Some alert fields contain data arrays. For instance, these include alert fields containing related events, assets, and user 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** parameter 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 HTML tag <strong>:

```
<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 strings 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, you must specify the path {{ A.C }} to query field 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<br> 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. Below is an 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.

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 }}
```

Connectors

Connectors are used for establishing connections between KUMA services and receiving events actively and passively.

The program has the following connector types available:

- tcp—used to receive data over TCP passively. Available for Windows and Linux agents.
- udp—used to receive data over UDP passively. Available for Windows and Linux agents.

- netflow—used to passively receive events in the NetFlow format.
- sflow—used to passively receive events in the SFlow format.
- nats-jetstream—used for communication with the NATS message broker. Available for Windows and Linux agents.
- kafka—used for communication with the Apache Kafka data bus. Available for Windows and Linux agents.
- http—used for receiving events over HTTP. Available for Windows and Linux agents.
- sql—used for selecting data from a database.

  The program supports the following types of SQL databases:

  - SQLite.
  - MSSQL.
  - MySQL.
  - PostgreSQL.
  - Cockroach.
  - Oracle.
  - Firebird.
- file—used to retrieve data from a text file. Available for Linux agents.
- 1c-log and 1c-xml are used to receive data from 1C logs. Available for Linux agents.
- diode—used for unidirectional data transfer in industrial ICS networks using data diodes.
- ftp—used to receive data over the File Transfer Protocol. Available for Windows and Linux agents.
- nfs—used to receive data over the Network File System protocol. Available for Windows and Linux agents.
- wmi—used to obtain data using Windows Management Instrumentation. Available for Windows agents.
- wec—used to receive data using Windows Event Forwarding (WEF) and Windows Event Collector (WEC), or local operating system logs of a Windows host. Available for Windows agents.
- snmp—used to receive data using the Simple Network Management Protocol. Available for Windows and Linux agents.
- snmp-trap—used to receive data using Simple Network Management Protocol traps (SNMP traps). Available for Windows and Linux agents.
- 'kata/edr' is used to receive KEDR data via the API.
- 'vmware' is used to receive VMware vCenter data via the API.

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.

Tcp type

When creating this type of connector, you need to define values for the following settings:

Basic settings tab:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—connector type, **tcp**.
- **URL** (required)—URL that you need to connect to. Available formats: hostname:port, IPv4:port, IPv6:port, :port.
- **Delimiter** is used to specify a character representing the delimiter between events. Available values: `\n`, `\t`, `\0`. If no separator is specified (an empty value is selected), the default value is `\n`.
- **Description**—resource description: up to 4,000 Unicode characters.

Advanced settings tab:

- **Buffer size** is used to set a buffer size for the connector. The default value is 1 MB, and the maximum value is 64 MB.
- **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
- **TLS mode**—TLS encryption mode using certificates in PEM x509 format:
  - **Disabled** (default)—do not use TLS encryption.
  - **Enabled**—use encryption without certificate verification.
  - **With verification**—use encryption with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during program installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/.
  - **Custom PFX** – use encryption. When this option is selected, a certificate must be generated with a private key in PKCS#12 container format in an external Certificate Authority. Then the certificate must be exported from the key store and uploaded to the KUMA web interface as a PFX secret. **Add PFX secret**.

    1. If you previously uploaded a PFX certificate, select it from the **Secret** drop-down list.

       If no certificate was previously added, the drop-down list shows **No data**.

    2. If you want to add a new certificate, click the + button on the right of the **Secret** list.

       The **Secret** window opens.

    3. In the **Name** field, enter the name that will be used to display the secret in the list of available secrets.
    4. Click the **Upload PFX** button to select the file containing your previously exported certificate with a private key in PKCS#12 container format.
    5. In the **Password** field, enter the certificate security password that was set in the Certificate Export Wizard.
    6. Click the **Save** button.

    The certificate will be added and displayed in the **Secret** list.

When using TLS, it is impossible to specify an IP address as a URL.

- **Compression**—you can use Snappy compression. By default, compression is **disabled**.
- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

udp type

When creating this type of connector, you need to define values for the following settings:

Basic settings tab:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—connector type, **udp**.
- **URL** (required)—URL that you need to connect to. Available formats: hostname:port, IPv4:port, IPv6:port, :port.
- **Delimiter** is used to specify a character representing the delimiter between events. Available values: `\n`, `\t`, `\0`. If no separator is specified (an empty value is selected), events are not separated.
- **Description**—resource description: up to 4,000 Unicode characters.

Advanced settings tab:

- **Buffer size** is used to set a buffer size for the connector. The default value is 16 KB, and the maximum value is 64 KB.
- **Workers**—used to set worker count for the connector. The default value is 1.
- **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
- **Compression**—you can use Snappy compression. By default, compression is **disabled**.
- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Netflow type

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.
  - **Type** (required)—connector type, **netflow**.

  - **URL** (required)—URL that you need to connect to.
  - **Description**—resource description: up to 4,000 Unicode characters.
- **Advanced settings** tab:
  - **Buffer size** is used to set a buffer size for the connector. The default value is 16 KB, and the maximum value is 64 KB.
  - **Workers**—used to set worker count for the connector. The default value is 1.
  - **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Sflow type

When creating this type of connector, you need to define values for the following settings:

Basic settings tab:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—connector type, **sflow**.
- **URL** (required)—a URL that you need to connect to. Available formats: hostname:port, IPv4:port, IPv6:port, :port.
- **Description**—resource description: up to 4,000 Unicode characters.

Advanced settings tab:

- **Buffer size** is used to set a buffer size for the connector. The default value is 1 MB, and the maximum value is 64 MB.
- **Workers**—used to set the amount of workers for a connector. The default value is 1.
- **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
- **Debug**—a toggle switch that lets you enable resource logging. By default, this toggle switch is in the **Disabled** position.

nats-jetstream type

When creating this type of connector, you need to define values for the following settings:

Basic settings tab:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.

- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—connector type, **nats-jetstream**.
- **URL** (required)—URL that you need to connect to.
- **Topic** (required)—the topic for NATS messages. Must contain Unicode characters.
- **Delimiter** is used to specify a character representing the delimiter between events. Available values: `\n`, `\t`, `\0`. If no separator is specified (an empty value is selected), events are not separated.
- **Description**—resource description: up to 4,000 Unicode characters.

Advanced settings tab:

- **Buffer size** is used to set a buffer size for the connector. The default value is 16 KB, and the maximum value is 64 KB.
- **GroupID**—the GroupID parameter for NATS messages. Must contain 1 to 255 Unicode characters. The default value is `default`.
- **Workers**—used to set worker count for the connector. The default value is 1.
- **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
- **Cluster ID** is the ID of the NATS cluster.
- **TLS mode** specifies whether TLS encryption is used:
  - **Disabled** (default)—do not use TLS encryption.
  - **Enabled**—use encryption without certificate verification.
  - **With verification**—use encryption with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during program installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/.
  - **Custom CA**—use encryption with verification that the certificate was signed by a Certificate Authority. The secret containing the certificate is selected from the **Custom CA** drop-down list, which is displayed when this option is selected.

    Creating a certificate signed by a Certificate Authority

To use this TLS mode, you must do the following on the KUMA Core server (OpenSSL commands are used in the examples below):

1. Create the key that will be used by the Certificate Authority.

   Example command:

   ```
   openssl genrsa -out ca.key 2048
   ```

2. Generate a certificate for the key that was just created.

   Example command:

   ```
   openssl req -new -x509 -days 365 -key ca.key -subj "/CN=<common host name of Certificate Authority>" -out ca.crt
   ```

3. Create a private key and a request to have it signed by the Certificate Authority.

   Example command:

   ```
   openssl req -newkey rsa:2048 -nodes -keyout server.key -subj "/CN=<common host name of KUMA server>" -out server.csr
   ```

4. Create a certificate signed by the Certificate Authority. The subjectAltName must include the domain names or IP addresses of the server for which the certificate is being created.

   Example command:

   ```
   openssl x509 -req -extfile <(printf "subjectAltName=DNS:domain1.ru,DNS:domain2.com,IP:192.168.0.1") -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt
   ```

5. The obtained `server.crt` certificate should be uploaded in the KUMA web interface as a **certificate**-type secret, which should then be selected from the **Custom CA** drop-down list.

When using TLS, it is impossible to specify an IP address as a URL.

To use KUMA certificates on third-party devices, you must change the certificate file extension from CERT to CRT. Otherwise, error x509: certificate signed by unknown authority may be returned.

- **Compression**—you can use Snappy compression. By default, compression is **disabled**.
- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Kafka type

When creating this type of connector, you need to define values for the following settings:

Basic settings tab:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—connector type, **kafka**.

- **URL**—URL that you need to connect to. Available formats: hostname:port, IPv4:port, IPv6:port.

- **Topic**—subject of Kafka messages. Must contain from 1 to 255 of the following characters: a–z, A–Z, 0–9, ".", "_", "-".

- **Authorization**—requirement for Agents to complete authorization when connecting to the connector:

  - **disabled** (by default).

  - PFX.

    When this option is selected, a certificate must be generated with a private key in PKCS#12 container format in an external Certificate Authority. Then the certificate must be exported from the key store and uploaded to the KUMA web interface as a PFX secret.

    Add PFX secret

    1. If you previously uploaded a PFX certificate, select it from the **Secret** drop-down list.

       If no certificate was previously added, the drop-down list shows **No data**.

    2. If you want to add a new certificate, click the + button on the right of the **Secret** list.

       The **Secret** window opens.

    3. In the **Name** field, enter the name that will be used to display the secret in the list of available secrets.

    4. Click the **Upload PFX** button to select the file containing your previously exported certificate with a private key in PKCS#12 container format.

    5. In the **Password** field, enter the certificate security password that was set in the Certificate Export Wizard.

    6. Click the **Save** button.

    The certificate will be added and displayed in the **Secret** list.

  - **plain**.

    If this option is selected, you must indicate the secret containing user account credentials for authorization when connecting to the connector.

    Add secret

1. If you previously created a secret, select it from the **Secret** drop-down list.

   If no secret was previously added, the drop-down list shows **No data**.

2. If you want to add a new secret, click the + button on the right of the **Secret** list.

   The **Secret** window opens.

3. In the **Name** field, enter the name that will be used to display the secret in the list of available secrets.

4. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector.

5. If necessary, add any other information about the secret in the **Description** field.

6. Click the **Save** button.

The secret will be added and displayed in the **Secret** list.

- **GroupID**—the GroupID parameter for Kafka messages. Must contain from 1 to 255 of the following characters: a–z, A–Z, 0–9, ".", "_", "-".

- **Description**—resource description: up to 4,000 Unicode characters.

Advanced settings tab:

- **Size of message to fetch**—should be specified in bytes. The default value is 16 MB.

- **Maximum fetch wait time**—timeout for a message of the defined size. The default value is 5 seconds.

- **Character encoding** setting specifies character encoding. The default value is `UTF-8`.

- **TLS mode** specifies whether TLS encryption is used:

  - **Disabled** (default)—do not use TLS encryption.

  - **Enabled**—use encryption without certificate verification.

  - **With verification**—use encryption with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during program installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/.

  - **Custom CA**—use encryption with verification that the certificate was signed by a Certificate Authority. The secret containing the certificate is selected from the **Custom CA** drop-down list, which is displayed when this option is selected.

    Creating a certificate signed by a Certificate Authority

To use this TLS mode, you must do the following on the KUMA Core server (OpenSSL commands are used in the examples below):

1. Create the key that will be used by the Certificate Authority.

   Example command:

   ```
   openssl genrsa -out ca.key 2048
   ```

2. Generate a certificate for the key that was just created.

   Example command:

   ```
   openssl req -new -x509 -days 365 -key ca.key -subj "/CN=<common host name of
   Certificate Authority>" -out ca.crt
   ```

3. Create a private key and a request to have it signed by the Certificate Authority.

   Example command:

   ```
   openssl req -newkey rsa:2048 -nodes -keyout server.key -subj "/CN=<common host
   name of KUMA server>" -out server.csr
   ```

4. Create a certificate signed by the Certificate Authority. The subjectAltName must include the domain names or IP addresses of the server for which the certificate is being created.

   Example command:

   ```
   openssl x509 -req -extfile <(printf
   "subjectAltName=DNS:domain1.ru,DNS:domain2.com,IP:192.168.0.1") -days 365 -in
   server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt
   ```

5. The obtained `server.crt` certificate should be uploaded in the KUMA web interface as a **certificate**-type secret, which should then be selected from the **Custom CA** drop-down list.

When using TLS, it is impossible to specify an IP address as a URL.

To use KUMA certificates on third-party devices, you must change the certificate file extension from CERT to CRT. Otherwise, error x509: certificate signed by unknown authority may be returned.

- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

kata/edr type

When creating this type of connector, you need to define values for the following settings:

Basic settings tab:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—connector type, **kata/edr**.
- **URL** (required)—URL at which telemetry can be received from the KATA/EDR server. The URL must include the host and the port; the default port is 443. If KATA/EDR is deployed in a cluster, you can specify multiple URLs to

ensure high availability of the connection.

- **Secret** (required) is a drop-down list in which you can select the secret which stores the credentials for connecting to the KATA/EDR server. You can select the secret resource from the drop-down list or create one using the + button. When creating a secret, you can specify a custom certificate and private key, or automatically generate a new self-signed certificate and private key. You can change the selected secret by clicking ✎.
- **External ID**—ID for external systems. KUMA generates an ID in this field automatically.
- **Description**—resource description: up to 4,000 Unicode characters.

Advanced settings tab:

- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.
- **Character encoding**—the source character encoding setting for conversion to UTF-8. We only recommend configuring a conversion if you find invalid characters in the fields of the normalized event. Default value: not selected.
- **Number of events**—maximum number of events in one request. By default, the value set on the KATA/EDR server is used.
- **Events fetch timeout** is the time in seconds to wait for receipt of events from the KATA/EDR server. The default value is 0, which means that the value specified on the KATA/EDR server is used.
- **Client timeout** is the time in seconds to wait for a response from the KATA/EDR server. Default value: 1,800 s; displayed as 0.
- **KEDRQL filter**—filter of requests to the KATA/EDR server. For more details on the query language, please refer to the KEDR Help.

Http type

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.
  - **Type** (required)—connector type, **http**.
  - **URL** (required)—URL that you need to connect to. Available formats: hostname:port, IPv4:port, IPv6:port, :port.
  - **Delimiter** is used to specify a character representing the delimiter between events. Available values: `\n`, `\t`, `\0`. If no separator is specified (an empty value is selected), events are not separated.
  - **Description**—resource description: up to 4,000 Unicode characters.
- **Advanced settings** tab:

- **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
- **TLS mode** specifies whether TLS encryption is used:
  - **Disabled** (default)—do not use TLS encryption.
  - **Enabled**—encryption is enabled, but without verification.
  - **With verification**—use encryption with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during program installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/.

  When using TLS, it is impossible to specify an IP address as a URL.
- **Proxy**—a drop-down list where you can select a proxy server resource.
- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Sql type

KUMA supports multiple **types of databases**.

The program supports the following types of SQL databases:

- SQLite.
- MSSQL.
- MySQL.
- PostgreSQL.
- Cockroach.
- Oracle.
- Firebird.

When creating a connector, you must specify general connector settings and specific database connection settings.

On the **Basic settings** tab, you must specify the following settings for the connector:

- **Name** (required)—unique name of the resource. Must contain 1 to 128 Unicode characters.
- **Type** (required)—connector type, **sql**.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Default query** (required)—SQL query that is executed when connecting to the database.

- **Reconnect to the database every time a query is sent** — the check box is cleared by default.
- **Poll interval, sec** —interval for executing SQL queries. This value is specified in seconds. The default value is 10 seconds.
- **Description**—resource description: up to 4,000 Unicode characters.

To connect to the database, you must define the values of the following settings on the **Basic settings** tab:

- **URL** (required)—secret that stores a list of URLs for connecting to the database.

  If necessary, you can **edit** or **create a secret**.

1. Click the + button.

   The secret window is displayed.

2. Define the values for the following settings:

   a. **Name**—the name of the added secret.

   b. **Type**—**urls**.

      This value is set by default and cannot be changed.

   c. **URL**—URL of the database.

      You must keep in mind that each type of database uses its own URL format for connections.

      Available URL formats are as follows:

      - For SQLite:

        - `sqlite3://file:<file_path>`

        > A question mark (?) is used as a placeholder.

      - For MSSQL:

        - `sqlserver://<user>:<password>@<server:port>/<instance_name>?database=<database>` (recommended)

        - `sqlserver://<user>:<password>@<server>?database=<database>&encrypt=disable`

        > The characters @p1 are used as a placeholder.

      - For MySQL:

        - `mysql://<user>:<password>@tcp(<server>:<port>)/<database>`

        > The characters %s are used as a placeholder.

      - For PostgreSQL:

        - `postgres://<user>:<password>@<server>/<database>?sslmode=disable`

        > The characters $1 are used as a placeholder.

      - For Cockroach:

        - `postgres://<user>:<password>@<server>:<port>/<database>?sslmode=disable`

The characters $1 are used as a placeholder.

- For Firebird:
  - `firebirdsql://<user>:<password>@<server>:<port>/<database>`

A question mark (?) is used as a placeholder.

d. **Description**—any additional information.

3. If necessary, click **Add** and specify an additional URL.

   In this case, if one URL is not available, the program connects to the next URL specified in the list of addresses.

4. Click the **Save** button.

1. Click the ✎ button.

   The secret window is displayed.

2. Specify the values for the settings that you want to change.

   You can change the following values:

   a. **Name**—the name of the added secret.

   b. **URL**—URL of the database.

      You must keep in mind that each type of database uses its own URL format for connections.

      Available URL formats are as follows:

      - For SQLite:

        - `sqlite3://file:<file_path>`

        A question mark (?) is used as a placeholder.

      - For MSSQL:

        - `sqlserver://<user>:<password>@<server:port>/<instance_name>?database=<database>` (recommended)

        - `sqlserver://<user>:<password>@<server>?database=<database>&encrypt=disable`

        The characters @p1 are used as a placeholder.

      - For MySQL:

        - `mysql://<user>:<password>@tcp(<server>:<port>)/<database>`

        The characters `?` are used as placeholders.

      - For PostgreSQL:

        - `postgres://<user>:<password>@<server>/<database>?sslmode=disable`

        The characters $1 are used as a placeholder.

      - For Cockroach:

        - `postgres://<user>:<password>@<server>:<port>/<database>?sslmode=disable`

        The characters $1 are used as a placeholder.

- For Firebird:
    - `firebirdsql://<user>:<password>@<server>:<port>/<database>`

  A question mark (?) is used as a placeholder.

  c. **Description**—any additional information.

3. If necessary, click **Add** and specify an additional URL.

   In this case, if one URL is not available, the program connects to the next URL specified in the list of addresses.

4. Click the **Save** button.

When creating connections, strings containing account credentials with special characters may be incorrectly processed. If an error occurs when creating a connection but you are sure that the settings are correct, enter the special characters in percent encoding.

Codes of special characters

!	#	$	%	&	'	(	)	*	+
%21	%23	%24	%25	%26	%27	%28	%29	%2A	%2B
,	/	:	;	=	?	@	[	]	\
%2C	%2F	%3A	%3B	%3D	%3F	%40	%5B	%5D	%5C

The following special characters are not supported in passwords used to access SQL databases: space, [,], :, /, #, %, \.

- **Identity column** (required)—name of the column that contains the ID for each row of the table.
- **Identity seed** (required)—identity column value that will be used to determine the specific line to start reading data from the SQL table.
- **Query**—field for an additional SQL query. The query indicated in this field is performed instead of the default query.
- **Poll interval, sec** —interval for executing SQL queries. The interval defined in this field replaces the default interval for the connector.

  This value is specified in seconds. The default value is 10 seconds.

On the **Advanced settings** tab, you need to specify the following settings for the connector:

- **Character encoding**—the specific encoding of the characters. The default value is `UTF-8`.

KUMA converts SQL responses to UTF-8 encoding. You can configure the SQL server to send responses in UTF-8 encoding or change the encoding of incoming messages on the KUMA side.

- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Within a single connector, you can **create a connection** for multiple supported databases.

To create a connection for multiple SQL databases:

1. Click the **Add connection** button.
2. Specify the **URL**, **Identity column**, **Identity seed**, **Query**, and **Poll interval, sec** values.
3. Repeat steps 1–2 for each required connection.

If a collector with a connector of the sql type cannot be started, check if the /opt/kaspersky/kuma/collector/<`collector ID`>/sql/state-<`file ID`> state file is empty.

If that state file is empty, delete it and restart the collector.

Supported SQL types and their specific usage features

The UNION operator is not supported by the SQL Connector resources.

The following SQL types are supported:

- MSSQL

  Example URLs:

  - `sqlserver://{user}:{password}@{server:port}/{instance_name}?database={database}` – (recommended option)
  - `sqlserver://{user}:{password}@{server}?database={database}`

  The characters `@p1` are used as a placeholder in the SQL query.

  If you need to connect using domain account credentials, specify the account name in `<domain>%5C<user>` format. For example: `sqlserver://domain%5Cuser:password@ksc.example.com:1433/SQLEXPRESS?database=KAV`.

- MySQL

  Example URL: `mysql://{user}:{password}@tcp({server}:{port})/{database}`

  The characters `?` are used as placeholders in the SQL query.

- PostgreSQL

  Example URL: `postgres://{user}:{password}@{server}/{database}?sslmode=disable`

  The characters `$1` are used as a placeholder in the SQL query.

- CockroachDB

  Example URL: `postgres://{user}:{password}@{server}:{port}/{database}?sslmode=disable`

The characters `$1` are used as a placeholder in the SQL query.

- SQLite3

  Example URL: `sqlite3://file:{file_path}`

  A question mark (`?`) is used as a placeholder in the SQL query.

  When querying SQLite3, if the initial value of the ID is in datetime format, you must add a date conversion with the sqlite datetime function to the SQL query. For example: select * from connections where datetime(login_time) > datetime(?, 'utc') order by login_time. In this example, `connections` is the SQLite table, and the value of the variable `?` is taken from the **Identity seed** field, and it must be specified in the {date}T{time}Z format (for example, 2021-01-01T00:10:00Z).

- Oracle DB

  In version 2.1.3 or later, KUMA uses a new driver for connecting to oracle. When upgrading, KUMA renames the connection secret to 'oracle-deprecated' and the connector continues to work. If no events are received after starting the collector with the 'oracle-deprecated' driver type, create a new secret with the 'oracle' driver and use it for connecting.

  We recommend using the new driver.

  Example URL of a secret with the new 'oracle' driver:

  oracle://{user}:{password}@{server}:{port}/{service_name}

  oracle://{user}:{password}@{server}:{port}/?SID={SID_VALUE}

  Example URL of a secret with the legacy 'oracle-deprecated' driver:

  oracle-deprecated://{user}/{password}@{server}:{port}/{service_name}

  The `:val` SQL variable is used as a placeholder in.

  When accessing Oracle DB, if the initial ID value is used in the datetime format, you must consider the type of the field in the database itself and, if necessary, add conversions of the time string in the query to ensure correct operation of the sql connector. For example, if the Connections table in the database has a login_time field, the following conversions are possible:

  - If the login_time field has the TIMESTAMP type, then depending on the database settings, the login_time field may contain a value in the YYYY-MM-DD HH24:MI:SS format (for example, 2021-01-01 00:00:00). Then, in the **Identity seed** field, specify 2021-01-01T00:00:00Z, and perform the conversion in the query using the to_timestamp function. For example:

    ```
    select * from connections where login_time > to_timestamp(:val, 'YYYY-MM-DD"T"HH24:MI:SS"Z"')
    ```

  - If the login_time field has the TIMESTAMP type, then depending on the database settings, the login_time field may contain a value in the YYYY-MM-DD"T"HH24:MI:SSTZH:TZM format (for example, 2021-01-01T00:00:00+03:00). Then, in the **Identity seed** field, specify 2021-01-01T00:00:00+03:00, and perform the conversion in the query using the to_timestamp_tz function. For example:

    ```
    select * from connections_tz where login_time > to_timestamp_tz(:val, 'YYYY-MM-DD"T"HH24:MI:SSTZH:TZM')
    ```

    For more details about the to_timestamp and to_timestamp_tz functions, refer to the official Oracle documentation.

  > To interact with Oracle DB, you must install the libaio1 Astra Linux package.

- Firebird® SQL

  Example URL:

`firebirdsql://{user}:{password}@{server}:{port}/{database}`

A question mark (`?`) is used as a placeholder in the SQL query.

If a problem occurs when connecting Firebird on Windows, use the full path to the database file. For example:

`firebirdsql://{user}:{password}@{server}:{port}/C:\Users\user\firebird\db.FDB`

A sequential request for database information is supported in SQL queries. For example, if you type `select * from <name of data table> where id > <placeholder>` in the **Query** field, the **Identity seed** field value will be used as the placeholder value the first time you query the table. In addition, the service that utilizes the SQL connector saves the ID of the last read entry, and the ID of this entry will be used as the placeholder value in the next query to the database.

Examples of SQL requests

SQLite, Firebird—`select * from table_name where id > ?`

MSSQL—`select * from table_name where id > @p1`

MySQL—`select * from table_name where id > ?`

PostgreSQL, Cockroach—`select * from table_name where id > $1`

Oracle—`select * from table_name where id > :val`

File type

The **file** type is used to retrieve data from any text file. One string in a file is considered to be one event. Strings delimiter: \n. This type of connector is available for Linux Agents.

To set up file transfers from a Windows server for processing by the KUMA collector:

1. On the Windows server, grant read access over the network to the folder with the files that you want processed.

2. On the Linux server, mount the shared folder on the Windows server (see the list of supported operating systems).

3. On the Linux server, install the collector that you want to process files from the mounted shared folder.

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.
  - **Type** (required)—connector type, **file**.
  - **URL** (required)—full path to the file that you need to interact with. For example, `/var/log/*som?[1-9].log`.

    File and folder mask templates

Masks:

- '*'—matches any sequence of characters.
- '[' ['^'] { range of characters } ']'—class of characters (should not be left blank).
- '?'—matches any single character.

Ranges of characters:

- [0-9]—digits;
- [a-zA-Z]—Latin alphabet characters.

Examples:

- /var/log/*som?[1-9].log
- /mnt/dns_logs/*/dns.log
- /mnt/proxy/access*.log

Limitations when using prefixes in file paths ?

Prefixes that cannot be used when specifying paths to files:

- /*
- /bin
- /boot
- /dev
- /etc
- /home
- /lib
- /lib64
- /proc
- /root
- /run
- /sys
- /tmp
- /usr/*
- /usr/bin/
- /usr/local/*
- /usr/local/sbin/
- /usr/local/bin/
- /usr/sbin/
- /usr/lib/
- /usr/lib64/
- /var/*
- /var/lib/
- /var/run/
- /opt/kaspersky/kuma/

Files are available at the following paths:

- /opt/kaspersky/kuma/clickhouse/logs/

- /opt/kaspersky/kuma/mongodb/log/
- /opt/kaspersky/kuma/victoria-metrics/log/

Limiting the number of files for watching by mask

The number of files simultaneously watched by mask can be limited by the max_user_watches setting of the Core. To view the value of a setting, run the following command:

```
cat /proc/sys/fs/inotify/max_user_watches
```

If the number of files for watching exceeds the value of the max_user_watches setting, the collector cannot read any more events from the files and the following error is written to the collector log:

```
Failed to add files for watching {"error": "no space left on device"}
```

To make sure that the collector continues to work correctly, you can configure the appropriate rotation of files so that the number of files does not exceed the value of the max_user_watches setting, or increase the max_user_watches value.

To increase the value of the setting:

```
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.

  - **Description**—resource description: up to 4,000 Unicode characters.
- **Advanced settings** tab:
  - **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Type 1c-xml

The **1c-xml** type is used to retrieve data from 1C application registration logs. When the connector handles multi-line events, it converts them into single-line events. This type of connector is available for Linux Agents.

When creating this type of connector, specify values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.

- **Type** (required)—connector type, **1c-xml**.
- **URL** (required)—full path to the directory containing files that you need to interact with. For example, `/var/log/1c/logs/`.

  Limitations when using prefixes in file paths ?

Prefixes that cannot be used when specifying paths to files:

- /*
- /bin
- /boot
- /dev
- /etc
- /home
- /lib
- /lib64
- /proc
- /root
- /run
- /sys
- /tmp
- /usr/*
- /usr/bin/
- /usr/local/*
- /usr/local/sbin/
- /usr/local/bin/
- /usr/sbin/
- /usr/lib/
- /usr/lib64/
- /var/*
- /var/lib/
- /var/run/
- /opt/kaspersky/kuma/

Files are available at the following paths:

- /opt/kaspersky/kuma/clickhouse/logs/

- /opt/kaspersky/kuma/mongodb/log/
- /opt/kaspersky/kuma/victoria-metrics/log/

- **Description**—resource description: up to 4,000 Unicode characters.

- **Advanced settings** tab:
  - **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Connector operation diagram:

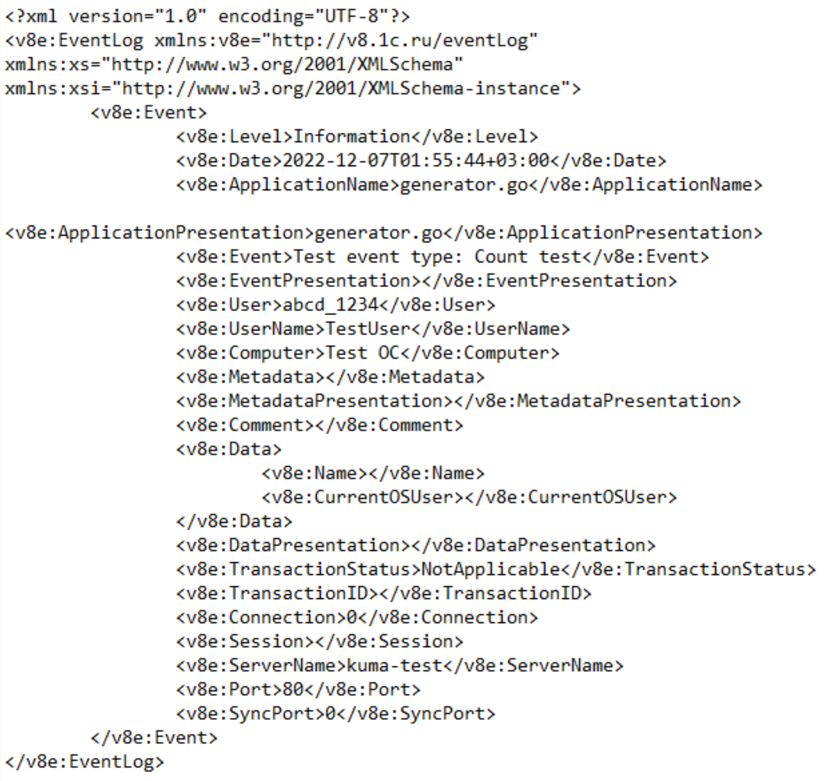
1. The files containing 1C logs with the XML extension are searched within the specified directory. Logs are placed in the directory either manually or using an application written in the 1C language, for example, using the ВыгрузитьЖурналРегистрации() function. The connector only supports logs received this way. For more information on how to obtain 1C logs, see the official 1C documentation.
2. Files are sorted by the last modification time in ascending order. All the files modified before the last read are discarded.

   Information about processed files is stored in the file /<collector working directory>/1c_xml_connector/state.ini and has the following format: "offset=<number>\ndev=<number>\ninode=<number>".
3. Events are defined in each unread file.
4. Events from the file are processed one by one. Multi-line events are converted to single-line events.

Connector limitations:

- Installation of a collector with a 1c-xml connector is not supported in a Windows operating system. To set up file transfers of 1C log files for processing by the KUMA collector:
  1. On the Windows server, grant read access over the network to the folder with the 1C log files.
  2. On the Linux server, mount the shared folder with the 1C log files on the Windows server (see the list of supported operating systems).
  3. On the Linux server, install the collector that you want to process 1C log files from the mounted shared folder.
- Files with an incorrect event format are not read. For example, if event tags in the file are in Russian, the collector does not read such events.

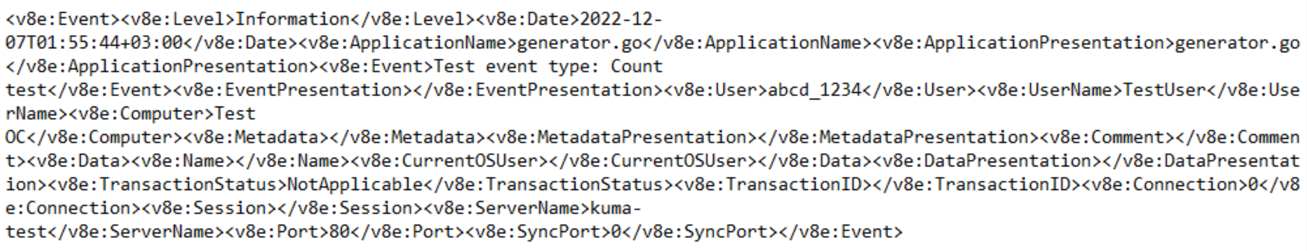
  Example of a correct XML file with an event ?.



```
<?xml version="1.0" encoding="UTF-8"?>
<v8e:EventLog xmlns:v8e="http://v8.1c.ru/eventLog"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <v8e:Event>
                <v8e:Level>Information</v8e:Level>
                <v8e:Date>2022-12-07T01:55:44+03:00</v8e:Date>
                <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:UserName>
                <v8e:Computer>Test OC</v8e:Computer>
                <v8e:Metadata></v8e:Metadata>
                <v8e:MetadataPresentation></v8e:MetadataPresentation>
                <v8e:Comment></v8e:Comment>
                <v8e:Data>
                        <v8e:Name></v8e:Name>
                        <v8e:CurrentOSUser></v8e:CurrentOSUser>
                </v8e:Data>
                <v8e:DataPresentation></v8e:DataPresentation>
                <v8e:TransactionStatus>NotApplicable</v8e:TransactionStatus>
                <v8e:TransactionID></v8e:TransactionID>
                <v8e:Connection>0</v8e:Connection>
                <v8e:Session></v8e:Session>
                <v8e:ServerName>kuma-test</v8e:ServerName>
                <v8e:Port>80</v8e:Port>
                <v8e:SyncPort>0</v8e:SyncPort>
        </v8e:Event>
</v8e:EventLog>
```

Example of a processed event ?.



```
<v8e:Event><v8e:Level>Information</v8e:Level><v8e:Date>2022-12-
07T01:55:44+03:00</v8e:Date><v8e:ApplicationName>generator.go</v8e:ApplicationName><v8e:ApplicationPresentation>generator.go
</v8e:ApplicationPresentation><v8e:Event>Test event type: Count
test</v8e:Event><v8e:EventPresentation></v8e:EventPresentation><v8e:User>abcd_1234</v8e:User><v8e:UserName>TestUser</v8e:Use
rName><v8e:Computer>Test
OC</v8e:Computer><v8e:Metadata></v8e:Metadata><v8e:MetadataPresentation></v8e:MetadataPresentation><v8e:Comment></v8e:Commen
t><v8e:Data><v8e:Name></v8e:Name><v8e:CurrentOSUser></v8e:CurrentOSUser></v8e:Data><v8e:DataPresentation></v8e:DataPresentat
ion><v8e:TransactionStatus>NotApplicable</v8e:TransactionStatus><v8e:TransactionID></v8e:TransactionID><v8e:Connection>0</v8
e:Connection><v8e:Session></v8e:Session><v8e:ServerName>kuma-
test</v8e:ServerName><v8e:Port>80</v8e:Port><v8e:SyncPort>0</v8e:SyncPort></v8e:Event>
```

- If a file read by the connector is enriched with the new events and if this file is not the last file read in the directory, all events from the file are processed again.

Type 1c-log

The **1c-log** type is used to retrieve data from 1C application technology logs. Strings delimiter: \n. The connector accepts only the first line from a multi-line event record. This type of connector is available for Linux Agents.

When creating this type of connector, specify values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.
  - **Type** (required)—connector type, **1c-log**.

- **URL** (required)—full path to the directory containing files that you need to interact with. For example, `/var/log/1c/logs/`.

  Limitations when using prefixes in file paths ?

Prefixes that cannot be used when specifying paths to files:

- /*
- /bin
- /boot
- /dev
- /etc
- /home
- /lib
- /lib64
- /proc
- /root
- /run
- /sys
- /tmp
- /usr/*
- /usr/bin/
- /usr/local/*
- /usr/local/sbin/
- /usr/local/bin/
- /usr/sbin/
- /usr/lib/
- /usr/lib64/
- /var/*
- /var/lib/
- /var/run/
- /opt/kaspersky/kuma/

Files are available at the following paths:

- /opt/kaspersky/kuma/clickhouse/logs/

- /opt/kaspersky/kuma/mongodb/log/
- /opt/kaspersky/kuma/victoria-metrics/log/

- **Description**—resource description: up to 4,000 Unicode characters.

- **Advanced settings** tab:
  - **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

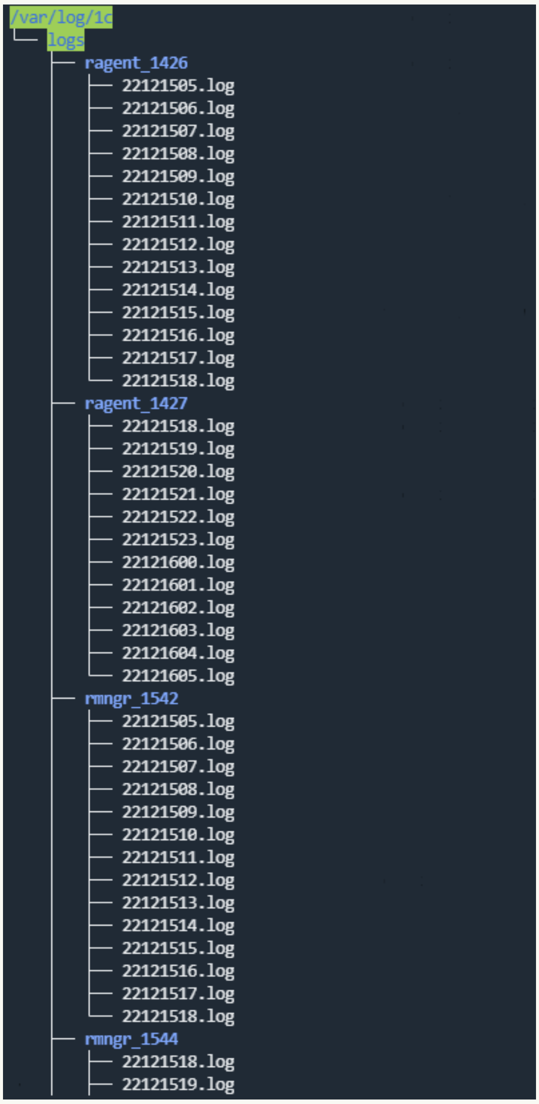
Connector operation diagram:

1. All 1C technology log files are searched.

   Log file requirements:

   - Files with the LOG extension are created in the log directory (`/var/log/1c/logs/` by default) within a subdirectory for each process.

     Example of a supported 1C technology log structure



```
/var/log/1c
└── logs
    ├── ragent_1426
    │   ├── 22121505.log
    │   ├── 22121506.log
    │   ├── 22121507.log
    │   ├── 22121508.log
    │   ├── 22121509.log
    │   ├── 22121510.log
    │   ├── 22121511.log
    │   ├── 22121512.log
    │   ├── 22121513.log
    │   ├── 22121514.log
    │   ├── 22121515.log
    │   ├── 22121516.log
    │   ├── 22121517.log
    │   └── 22121518.log
    ├── ragent_1427
    │   ├── 22121518.log
    │   ├── 22121519.log
    │   ├── 22121520.log
    │   ├── 22121521.log
    │   ├── 22121522.log
    │   ├── 22121523.log
    │   ├── 22121600.log
    │   ├── 22121601.log
    │   ├── 22121602.log
    │   ├── 22121603.log
    │   ├── 22121604.log
    │   └── 22121605.log
    ├── rmngr_1542
    │   ├── 22121505.log
    │   ├── 22121506.log
    │   ├── 22121507.log
    │   ├── 22121508.log
    │   ├── 22121509.log
    │   ├── 22121510.log
    │   ├── 22121511.log
    │   ├── 22121512.log
    │   ├── 22121513.log
    │   ├── 22121514.log
    │   ├── 22121515.log
    │   ├── 22121516.log
    │   ├── 22121517.log
    │   └── 22121518.log
    ├── rmngr_1544
    │   ├── 22121518.log
    │   ├── 22121519.log
```

   - Events are logged to a file for an hour; after that, the next log file is created.

  - The file names have the following format: `<YY><MM><DD><HH>.log`. For example, `22111418.log` is a file created in 2022, in the 11th month, on the 14th at 18:00.
  - Each event starts with the event time in the following format: <mm>:<ss>.<microseconds>-<duration_in_microseconds>.

2. The processed files are discarded.

   Information about processed files is stored in the file /<collector working directory>/1c_log_connector/state.json.

3. Processing of the new events starts, and the event time is converted to the RFC3339 format.

4. The next file in the queue is processed.

Connector limitations:

- Installation of a collector with a 1c-log connector is not supported in a Windows operating system. To set up file transfers of 1C log files for processing by the KUMA collector:
  1. On the Windows server, grant read access over the network to the folder with the 1C log files.
  2. On the Linux server, mount the shared folder with the 1C log files on the Windows server (see the list of supported operating systems).
  3. On the Linux server, install the collector that you want to process 1C log files files from the mounted shared folder.
- Only the first line from a multi-line event record is processed.
- The normalizer processes only the following types of events:
  - ADMIN
  - ATTN
  - CALL
  - CLSTR
  - CONN
  - DBMSSQL
  - DBMSSQLCONN
  - DBV8DBENG
  - EXCP
  - EXCPCNTX
  - HASP
  - LEAKS
  - LIC

- MEM
- PROC
- SCALL
- SCOM
- SDBL
- SESN
- SINTEG
- SRVC
- TLOCK
- TTIMEOUT
- VRSREQUEST
- VRSRESPONSE

Diode type

Used to transmit events using a data diode.

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.
  - **Type** (required)—connector type, **diode**.
  - **Data diode destination directory** (required)—full path to the KUMA collector server directory where the data diode moves files containing events from the isolated network segment. After the connector has read these files, the files are deleted from the directory. The path can contain up to 255 Unicode characters.

    Limitations when using prefixes in paths ?

Prefixes that cannot be used when specifying paths to files:

- /*
- /bin
- /boot
- /dev
- /etc
- /home
- /lib
- /lib64
- /proc
- /root
- /run
- /sys
- /tmp
- /usr/*
- /usr/bin/
- /usr/local/*
- /usr/local/sbin/
- /usr/local/bin/
- /usr/sbin/
- /usr/lib/
- /usr/lib64/
- /var/*
- /var/lib/
- /var/run/
- /opt/kaspersky/kuma/

Files are available at the following paths:

- /opt/kaspersky/kuma/clickhouse/logs/

- /opt/kaspersky/kuma/mongodb/log/
- /opt/kaspersky/kuma/victoria-metrics/log/

- **Delimiter** is used to specify a character representing the delimiter between events. Available values: `\n`, `\t`, `\0`. If no separator is specified (an empty value is selected), the default value is `\n`.

  This setting must match for the connector and destination resources used to relay events from an isolated network segment via the data diode.

- **Description**—resource description: up to 4,000 Unicode characters.

- **Advanced settings** tab:

  - **Workers**—the number of services processing the request queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.

  - **Poll interval, sec** —frequency at which the files are read from the directory containing events from the data diode. The default value is 2. The value is specified in seconds.

  - **Character encoding** setting specifies character encoding. The default value is `UTF-8`.

  - **Compression**—you can use Snappy compression. By default, compression is **disabled**.

    This setting must match for the connector and destination resources used to relay events from an isolated network segment via the data diode.

  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Ftp type

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:

  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.

  - **Tenant** (required)—name of the tenant that owns the resource.

  - **Type** (required)—connector type, **ftp**.

  - **URL** (required)—actual URL of the file or file mask beginning with 'ftp://'. For a file mask, you can use * ? [...].

    File mask templates

Masks:

- '*'—matches any sequence of characters.
- '[' ['^'] { range of characters } ']'—class of characters (should not be left blank).
- '?'—matches any single character.

Ranges of characters:

- [0-9]—digits;
- [a-zA-Z]—Latin alphabet characters.

Examples:

- /var/log/*som?[1-9].log
- /mnt/dns_logs/*/dns.log
- /mnt/proxy/access*.log

If the URL does not include the FTP server port, port 21 is inserted.

- **URL credentials**—for specifying the user name and password for the FTP server. If there is no user name and password, the line remains empty.
- **Description**—resource description: up to 4,000 Unicode characters.

- **Advanced settings** tab:
  - **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Nfs type

When creating this type of connector, you need to define values for the following settings:

Basic settings tab:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—connector type, **nfs**.
- **URL** (required)—path to the remote folder in the format nfs://host/path.
- **File name mask** (required)—mask used to filter files containing events. Use of masks is acceptable "`*`", "`?`", "`[...]`".

- **Poll interval, sec**—polling interval. The time interval after which files are re-read from the remote system. The value is specified in seconds. The default value is 0.
- **Description**—resource description: up to 4,000 Unicode characters.

Advanced settings tab:

- **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

vmware type

When creating this type of connector, you need to define values for the following settings:

Basic settings tab:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—connector type, **vmware**.
- **URL** (required)—the URL at which the VMware API is available. The URL must include the host and the port. Only one URL can be specified.
- **VMware credentials** (required) — a secret that stores the username and password for connecting to the VMware API.
- **Client timeout** is the time to wait after a request that did not return events before making a new request. This value is specified in seconds. The default value is 5 seconds. If value is 0, the default value is used.
- **Maximum number of events** — number of events requested from the VMware API in one request. The default value is 100. The maximum value is 1000.
- **Start timestamp**—starting date and time from which you want to read events from the VMware API. The default value is the time when the collector was started. If started after the collector is stopped, the events are read from the last saved date.

Advanced settings tab:

- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.
- **Character encoding**—specifies the character encoding. The default is UTF-8.
- **TLS mode**—TLS encryption mode using certificates in PEM x509 format:
  - **Disabled** (default)—do not use TLS encryption.
  - **Enabled** means encryption is used, but certificates are not verified.

    - **Custom CA**—this option requires you to add a secret with a certificate to the collector. Not self-signed certificate. The certificate of the server must be signed with the certificate specified in the collector settings.

- **Custom CA** (required if "Custom CA" is selected for the TLS mode setting) is the secret where the certificate will be stored.

Wmi type

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:
    - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
    - **Tenant** (required)—name of the tenant that owns the resource.
    - **Type** (required)—connector type, **wmi**.
    - **URL** (required)—URL of the collector being created, for example: `kuma-collector.example.com:7221`.

      The creation of a collector for receiving data using Windows Management Instrumentation results in the automatic creation of an agent that receives the necessary data on the remote device and forwards that data to the collector service. In the **URL**, you must specify the address of this collector. The URL is known in advance if you already know on which server you plan to install the service. However, this field can also be filled after the Installation Wizard is finished by copying the URL data from the **Resources** → **Active services** section.
    - **Description**—resource description: up to 4,000 Unicode characters.
    - **Default credentials**—drop-down list that does not require any value to be selected. The account credentials used to connect to hosts must be provided in the **Remote hosts** table (see below).
    - The **Remote hosts** table lists the remote Windows assets that you can connect to. Available columns:
        - **Host** (required) is the IP address or name of the device from which you want to receive data. For example, "machine-1".
        - **Domain** (required)—name of the domain in which the remote device resides. For example, "example.com".
        - **Log type**—drop-down list to select the name of the Windows logs that you need to retrieve. By default, only preconfigured logs are displayed in the list, but you can add custom logs to the list by typing their name in the **Windows logs** field and then pressing **ENTER**. KUMA service and resource configurations may require additional changes in order to process custom logs correctly.

          Logs that are available by default:
            - Application
            - ForwardedEvents
            - Security
            - System
            - HardwareEvents

> If a WMI connection uses at least one log with an incorrect name, the agent that uses the connector does not receive events from all the logs within this connection, even if the names of other logs are specified correctly. The WMI agent connections for which all log names are specified correctly will work properly.

    - **Secret**—account credentials for accessing a remote Windows asset with permissions to read the logs. If you leave this field blank, the credentials from the secret selected in the **Default credentials** drop-down list are used. The login in the secret must be specified without the domain. The domain value for access to the host is taken from the **Domain** column of the **Remote hosts** table.

      You can select the secret resource from the drop-down list or create one using the + button. The selected secret can be changed by clicking on the ✎ button.

- **Advanced settings** tab:

  - **Character encoding** setting specifies character encoding. The default value is `UTF-8`.

  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Receiving events from a remote device

Conditions for receiving events from a remote Windows device hosting a KUMA agent:

- To start the KUMA agent on the remote device, you must use an account with the “Log on as a service” permissions.

- To receive events from the KUMA agent, you must use an account with Event Log Readers permissions. For domain servers, one such user account can be created so that a group policy can be used to distribute its rights to read logs to all servers and workstations in the domain.

- TCP ports 135, 445, and 49152–65535 must be opened on the remote Windows devices.

- You must run the following services on the remote machines:

  - Remote Procedure Call (RPC)

  - RPC Endpoint Mapper

Wec type

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:

  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.

  - **Tenant** (required)—name of the tenant that owns the resource.

  - **Type** (required)—connector type, **wec**.

  - **URL** (required)—URL of the collector being created, for example: `kuma-collector.example.com:7221`.

The creation of a collector for receiving data using Windows Event Collector results in the automatic creation of an agent that receives the necessary data on the remote device and forwards that data to the collector service. In the **URL**, you must specify the address of this collector. The URL is known in advance if you already know on which server you plan to install the service. However, this field can also be filled after the Installation Wizard is finished by copying the URL data from the **Resources** → **Active services** section.

- **Description**—resource description: up to 4,000 Unicode characters.

- **Windows logs** (required)—Select the names of the Windows logs you want to retrieve from this drop-down list. By default, only preconfigured logs are displayed in the list, but you can add custom logs to the list by typing their name in the **Windows logs** field and then pressing **ENTER**. KUMA service and resource configurations may require additional changes in order to process custom logs correctly.

  Preconfigured logs:

  - Application
  - ForwardedEvents
  - Security
  - System
  - HardwareEvents

  > If the name of at least one log is specified incorrectly, the agent using the connector does not receive events from any log, even if the names of other logs are correct.

- **Advanced settings** tab:
  - **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

To start the KUMA agent on the remote device, you must use a service account with the “Log on as a service” permissions. To receive events from the operating system log, the service user account must also have Event Log Readers permissions.

You can create one user account with “Log on as a service” and “Event Log Readers” permissions, and then use a group policy to extend the rights of this account to read the logs to all servers and workstations in the domain.

We recommend that you disable interactive logon for the service account.

snmp type

> To process events received via SNMP, you must use json normalizer.

It is available for Windows and Linux Agents. Supported protocol versions:

- snmpV1
- snmpV2

- snmpV3

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:
    - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
    - **Tenant** (required)—name of the tenant that owns the resource.
    - **Type** (required)—connector type, **snmp**.
    - **SNMP version** (required)—This drop-down list allows you to select the version of the protocol to use.
    - **Host** (required)—hostname or its IP address. Available formats: hostname, IPv4, IPv6.
    - **Port** (required)—port for connecting to the host. Typically 161 or 162 are used.

    The **SNMP version**, **Host** and **Port** settings define one connection to a SNMP resource. You can create several such connections in one connector by adding new ones using the **SNMP resource** button. You can delete connections by using the 🗑 button.

    - **Secret** (required) is a drop-down list to select the secret which stores the credentials for connecting via the Simple Network Management Protocol. The secret type must match the SNMP version. If required, a secret can be created in the connector creation window using the + button. The selected secret can be changed by clicking on the ✎ button.
    - In the **Source data** table you can specify the rules for naming the received data, according to which OIDs, object identifiers, will be converted into keys with which the normalizer can interact. Available table columns:
        - **Parameter name** (required)—an arbitrary name for the data type. For example, "Site name" or "Site uptime".
        - **OID** (required)—a unique identifier that determines where to look for the required data at the event source. For example, "1.3.6.1.2.1.1.5".
        - **Key** (required)—a unique identifier returned in response to a request to the asset with the value of the requested setting. For example, "sysName". This key can be accessed when normalizing data.
    - **Description**—resource description: up to 4,000 Unicode characters.
- **Advanced settings** tab:
    - **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
    - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Snmp-trap type

The **snmp-trap** connector is used in agents and collectors to passively receive SNMP trap messages. The connector receives and prepares messages for normalization by mapping the SNMP object IDs to the temporary keys. Then the message is passed to the JSON normalizer, where the temporary keys are mapped to the KUMA fields and an event is generated.

> To process events received via SNMP, you must use json normalizer.

It is available for Windows and Linux Agents. Supported protocol versions:

- snmpV1
- snmpV2

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.
  - **Type** (required)—connector type, **snmp-trap**.
  - **SNMP version** (required)—in this drop-down list, select the version of the protocol to be used: **snmpV1** or **snmpV2**.

    > For example, Windows uses the **snmpV2** version by default.

  - **URL** (required) – URL where SNMP Trap messages will be expected. Available formats: hostname:port, IPv4:port, IPv6:port, :port.

  The **SNMP version** and **URL** parameters define one connection used to receive SNMP Traps. You can create several such connections in one connector by adding new ones using the **SNMP resource** button. You can delete connections by using the ✕ button.

  - In the **Source data** table, specify the rules for naming the received data, according to which OIDs (object identifiers) are converted to the keys with which the normalizer can interact.

    You can click **Apply OIDs for WinEventLog** to populate the table with mappings for OID values that arrive in WinEventLog logs. If more data needs to be determined and normalized in the incoming events, add to the table rows containing OID objects and their keys.

    Available table columns:

    - **Parameter name** —an arbitrary name for the data type. For example, "`Site name`" or "`Site uptime`".
    - **OID** (required)—a unique identifier that determines where to look for the required data at the event source. For example, `1.3.6.1.2.1.1.1`.
    - **Key** (required)—a unique identifier returned in response to a request to the asset with the value of the requested setting. For example, `sysDescr`. This key can be accessed when normalizing data.

    Data is processed according to the allow list principle: objects that are not specified in the table are not sent to the normalizer for further processing.

  - **Description**—resource description: up to 4,000 Unicode characters.
- **Advanced settings** tab:

- **Character encoding** setting specifies character encoding. The default value is `UTF-8`. When receiving snmp-trap events from Windows with Russian localization, if you encounter invalid characters in the event, we recommend changing the character encoding in the snmp-trap connector to Windows 1251.
- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Configuring the source of SNMP trap messages for Windows

Configuring a Windows device to send SNMP trap messages to the KUMA collector involves the following steps:

1. Configuring and starting the SNMP and SNMP trap services
2. Configuring the Event to Trap Translator service

Events from the source of SNMP trap messages must be received by the KUMA collector, which uses a connector of the snmp-trap type and a json normalizer.

Configuring and starting the SNMP and SNMP trap services
To configure and start the SNMP and SNMP trap services in Windows 10:

1. Open **Settings** → **Apps** → **Apps and features** → **Optional features** → **Add feature** → **Simple Network Management Protocol (SNMP)** and click **Install**.
2. Wait for the installation to complete and restart your computer.
3. Make sure that the SNMP service is running. If any of the following services are not running, enable them:
   - **Services** → **SNMP Service**.
   - **Services** → **SNMP Trap**.
4. Right-click **Services** → **SNMP Service**, and in the context menu select **Properties**. Specify the following settings:
   - On the **Log On** tab, select the **Local System account** check box.
   - On the **Agent** tab, fill in the **Contact** (for example, specify `User-win10`) and Location (for example, specify `detroit`) fields.
   - On the **Traps** tab:
     - In the **Community Name** field, enter **community public** and click **Add to list**.
     - In the **Trap destination** field, click **Add**, specify the IP address or host of the KUMA server on which the collector that waits for SNMP events is deployed, and click **Add**.
   - On the **Security** tab:
     - Select the **Send authentication trap** check box.
     - In the **Accepted community names** table, click **Add**, enter **Community Name public** and specify **READ WRITE** as the **Community rights**.
     - Select the **Accept SNMP packets from any hosts** check box.
5. Click **Apply** and confirm your selection.

6. Right click **Services** → **SNMP Service** and select **Restart**.

To configure and start the SNMP and SNMP trap services in Windows XP:

1. Open **Start** → **Control Panel** → **Add or Remove Programs** → **Add / Remove Windows Components** → **Management and Monitoring Tools** → **Details**.
2. Select **Simple Network Management Protocol** and **WMI SNMP Provider**, and then click **OK** → **Next**.
3. Wait for the installation to complete and restart your computer.
4. Make sure that the SNMP service is running. If any of the following services are not running, enable them by setting the **Startup type** to **Automatic**:
   - **Services** → **SNMP Service**.
   - **Services** → **SNMP Trap**.
5. Right-click **Services** → **SNMP Service**, and in the context menu select **Properties**. Specify the following settings:
   - On the **Log On** tab, select the **Local System account** check box.
   - On the **Agent** tab, fill in the **Contact** (for example, specify `User-win10`) and Location (for example, specify `detroit`) fields.
   - On the **Traps** tab:
     - In the **Community Name** field, enter **community public** and click **Add to list**.
     - In the **Trap destination** field, click **Add**, specify the IP address or host of the KUMA server on which the collector that waits for SNMP events is deployed, and click **Add**.
   - On the **Security** tab:
     - Select the **Send authentication trap** check box.
     - In the **Accepted community names** table, click **Add**, enter **Community Name public** and specify **READ WRITE** as the **Community rights**.
     - Select the **Accept SNMP packets from any hosts** check box.
6. Click **Apply** and confirm your selection.
7. Right click **Services** → **SNMP Service** and select **Restart**.

Changing the port for the SNMP trap service

You can change the SNMP trap service port if necessary.

To change the port of the SNMP trap service:

1. Open the C:\Windows\System32\drivers\etc folder.
2. Open the **services** file in Notepad as an administrator.

3. In the **service name** section of the file, specify the snmp-trap connector port added to the KUMA collector for the **SNMP trap** service.
4. Save the file.
5. Open the Control Panel and select **Administrative Tools** → **Services**.
6. Right-click **SNMP Service** and select **Restart**.

Configuring the Event to Trap Translator service

To configure the Event to Trap Translator service that translates Windows events to SNMP trap messages:

1. In the command line, type `evntwin` and press **Enter**.
2. Under **Configuration type**, select **Custom**, and click the **Edit** button.
3. In the **Event sources** group of settings, 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.

Predefined connectors

The connectors listed in the table below are included in the KUMA distribution kit.

Predefined connectors

Connector name	Comment
[OOTB] Continent SQL	Obtains events from the database of the Continent hardware and software encryption system. To use it, you must configure the settings of the corresponding secret type.
[OOTB] InfoWatch Trafic Monitor SQL	Obtains events from the database of the InfoWatch Traffic Monitor system. To use it, you must configure the settings of the corresponding secret type.
[OOTB] KSC MSSQL	Obtains events from the MS SQL database of the Kaspersky Security Center 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	Obtains audit events from the Oracle database.

SQL	To use it, you must configure the settings of the corresponding secret type.
[OOTB] SecretNet SQL	Obtains events from the SecretNet SQL database. To use it, you must configure the settings of the corresponding secret type.

Secrets

Secrets are used to securely store sensitive information such as user names and passwords that must be used by KUMA to interact with external services. If a secret stores account data such as user login and password, when the collector connects to the event source, the account specified in the secret may be blocked in accordance with the password policy configured in the event source system.

Secrets can be used in the following KUMA services and features:

- Collector (when using TLS encryption).
- Connector (when using TLS encryption).
- Destinations (when using TLS encryption or authorization).
- Proxy servers.

Available settings:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—the type of secret.

  When you select the type in the drop-down list, the parameters for configuring this secret type also appear. These parameters are described below.
- **Description**—up to 4,000 Unicode characters.

Depending on the secret type, different fields are available. You can select one of the following secret types:

- **credentials**—this type of secret is used to store account credentials required to connect to external services, such as SMTP servers. If you select this type of secret, you must fill in the **User** and **Password** fields. If the **Secret** resource uses the 'credentials' type to connect the collector to an event source, for example, a database management system, the account specified in the secret may be blocked in accordance with the password policy configured in the event source system.
- **token**—this secret type is used to store tokens for API requests. Tokens are used when connecting to IRP systems, for example. If you select this type of secret, you must fill in the **Token** field.
- **ktl**—this secret type is used to store Kaspersky Threat Intelligence Portal account credentials. If you select this type of secret, you must fill in the following fields:
  - **User** and **Password** (required fields)—user name and password of your Kaspersky Threat Intelligence Portal account.
  - **PFX file** (required)—lets you upload a Kaspersky Threat Intelligence Portal certificate key.

  - **PFX password** (required)—the password for accessing the Kaspersky Threat Intelligence Portal certificate key.

- **urls**—this secret type is used to store URLs for connecting to SQL databases and proxy servers. In the **Description** field, you must provide a description of the connection for which you are using the secret of **urls** type.

  You can specify URLs in the following formats: hostname:port, IPv4:port, IPv6:port, :port.

- **pfx**—this type of secret is used for importing a PFX file containing certificates. If you select this type of secret, you must fill in the following fields:

  - **PFX file** (required)—this is used to upload a PFX file. The file must contain a certificate and key. PFX files may include CA-signed certificates for server certificate verification.

  - **PFX password** (required)—this is used to enter the password for accessing the certificate key.

- **kata/edr**—this type of secret is used to store the certificate file and private key required when connecting to the Kaspersky Endpoint Detection and Response server. If you select this type of secret, you must upload the following files:

  - Certificate file—KUMA server certificate.

    The file must be in PEM format. You can upload only one certificate file.

  - Private key for encrypting the connection—KUMA server RSA key.

    The key must be without a password and with the PRIVATE KEY header. You can upload only one key file.

  > You can generate certificate and key files by clicking the ⤓ button.

- **snmpV1**—this type of secret is used to store the values of **Community** access (for example, `public` or `private`) that is required for interaction over the Simple Network Management Protocol.

- **snmpV3**—this type of secret is used for storing data required for interaction over the Simple Network Management Protocol. If you select this type of secret, you must fill in the following fields:

  - **User**—user name indicated without a domain.

  - **Security Level**—security level of the user.

    - **NoAuthNoPriv**—messages are forwarded without authentication and without ensuring confidentiality.

    - **AuthNoPriv**—messages are forwarded with authentication but without ensuring confidentiality.

    - **AuthPriv**—messages are forwarded with authentication and ensured confidentiality.

    You may see additional settings depending on the selected level.

  - **Password**—SNMP user authentication password. This field becomes available when the **AuthNoPriv** or **AuthPriv** security level is selected.

  - **Authentication Protocol**—the following protocols are available: MD5, SHA, SHA224, SHA256, SHA384, SHA512. This field becomes available when the **AuthNoPriv** or **AuthPriv** security level is selected.

  - **Privacy Protocol**—protocol used for encrypting messages. Available protocols: DES, AES. This field becomes available when the **AuthPriv** security level is selected.

- **Privacy password**—encryption password that was set when the SNMP user was created. This field becomes available when the **AuthPriv** security level is selected.

- **certificate**—this secret type is used for storing certificate files. Files are uploaded to a resource by clicking the **Upload certificate file** button. X.509 certificate public keys in Base64 are supported.

Predefined secrets

The secrets listed in the table below are included in the KUMA distribution kit.

Predefined secrets

Secret name	Description
[OOTB] Continent SQL connection	Stores confidential data and settings for connecting to the APKSh Kontinent database. To use it, you must specify the login name and password of the database.
[OOTB] KSC MSSQL connection	Stores confidential data and settings for connecting to the MS SQL database of Kaspersky Security Center (KSC). To use it, you must specify the login name and password of the database.
[OOTB] KSC MySQL Connection	Stores confidential data and settings for connecting to the MySQL database of Kaspersky Security Center (KSC). To use it, you must specify the login name and password of the database.
[OOTB] Oracle Audit Trail SQL Connection	Stores confidential data and settings for connecting to the Oracle database. To use it, you must specify the login name and password of the database.
[OOTB] SecretNet SQL connection	Stores confidential data and settings for connecting to the MS SQL database of the SecretNet system. To use it, you must specify the login name and password of the database.

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.

Segmentation rule settings

Segmentation rules are created in the **Resources** → **Segmentation rules** section of the KUMA web interface.

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)—type of the segmentation rule. Available values:
  - **By filter**—alerts are 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.

    - 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.
      - **<**—the left operand is less than the right operand.
      - **<=**—the left operand is less than or equal to the right operand.
      - **>**—the left operand is greater than the right operand.
      - **>=**—the left operand is greater than or equal to the right operand.
      - **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
      - **contains**—the left operand contains values of the right operand.
      - **startsWith**—the left operand starts with one of the values of the right operand.
      - **endsWith**—the left operand ends with one of the values of the right operand.
      - **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
      - **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.

  - **By identical fields**—an alert is created if the correlation event contains the event fields specified in the **Correlation rule identical fields** group of settings.

    The fields are added using the **Add field** button. You can delete the added fields by clicking the cross icon or the **Reset** button.

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 <Alert creation date>)`
> - `Network. Possible port scan (from 10.10.10.10 <Alert creation date>)`

  - **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.

- **Alert naming template** (required)—a template for naming the alerts created according to 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 event field value 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> <Alert creation date>)".

- **Description**—resource description: up to 4,000 Unicode characters.

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 rule** (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.

During the correlation process, when entries are deleted from context tables, service events are generated in the correlators. These events only exist in the correlators, and they are not redirected to other destinations. Service events are sent for processing by correlation rules of that correlator which uses the context table. Correlation rules can be configured to track these events so that they can be used to identify threats.

Service event fields for deleting an entry from a context table are described below.

Event field	Value or comment
`ID`	Event ID
`Timestamp`	Time when the expired entry was deleted
`Name`	`"context table record expired"`
`DeviceVendor`	`"Kaspersky"`
`DeviceProduct`	`"KUMA"`
`ServiceID`	Correlator ID
`ServiceName`	Correlator name
`DeviceExternalID`	Context table ID
`DevicePayloadID`	Key of the expired entry
`BaseEventCount`	Number of updates for the deleted entry, incremented by one

Viewing the list of context tables

To view the context table list of the correlator:

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 for which you want to view context tables, select **Go to context tables**.

   The **Correlator context tables** list is displayed.

The table contains the following data:

- **Name**—name of the context table.
- **Size on disk**—size of the context table.
- **Directory**—path to the context table on the KUMA correlator server.

Adding a context table

To add a context table:

1. In the KUMA web interface, select the **Resources** section.

2. In the **Resources** section, click **Context tables**.

3. In the **Context tables** window, click **Add**.

   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 **Description** field, provide any additional information.

   You can use up to 4,000 Unicode characters.

   This field is optional.

8. In the **Schema** section, specify which fields the context table has and the data types of the fields.

   Depending on the data type, a field may or may not be a key field. At least one field in the table must be a key field. The names of all fields must be unique.

   To add a table row, click **Add** and fill in the table fields:

   a. In the **Name** field, enter the name of the field. The maximum length is 128 characters.

   b. In the **Type** drop-down list, select the data type for the field.

      Possible field data types

Possible data types of context table fields

Field data type	Can be a key field	Comment
Integer	Yes	–
Floating point number	Yes	–
String	Yes	–
Boolean	Yes	–
Timestamp	Yes	For a field of this type, it is checked that the field value is greater than or equal to zero. No other operations are provided.
IP address	Yes	For a field of this type, it is checked that the field value corresponds to the IPv4, IPv6 format. No other operations are provided.
Integer list	No	–
Float list	No	–
List of strings	No	–
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).

9. Add the required number of context table rows.

After saving the context table, the schema cannot be changed.

10. Click the **Save** button.

The context table is added.

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.
   - **TTL**—the record added to the context table is stored in it for this duration. This value is specified in seconds.
   - **Description**—any additional information about the resource.
   - **Schema** is an ordered list of fields and their data types, with key fields marked. If the context table is not used in a correlation rule, you can edit the list of fields.

     If you want to edit the schema in a context table that is already being used in a correlation rule, follow the steps below.

   The **Tenant** field is not available for editing.

5. Click **Save**.

To edit the settings of the context table previously used by the correlator:

1. Export data from the table.

2. Copy and save the path to the file with the data of the table on the disk of the correlator. This path is specified in the **Directory** column in the **Correlator context tables** window. You will need this path later to delete the file from the disk of the correlator.

3. Delete the context table from the correlator.

4. Edit context table settings as necessary.

5. Delete the file with data of the table on the disk of the correlator at the path from step 2.

6. Add the context table in which you edited the settings to the correlator.

7. To restart the correlator, in the **Resources** → **Active services** section, in the list of services, select the check box next to the relevant correlator, click the three-dots icon on the toolbar and in the displayed menu, select **Restart**.

8. Adapt the fields in the exported table (see step 1) so that they match the fields of the table that you uploaded to the correlator at step 6.

9. Import the adapted data to the context table.

Duplicating context table settings

To copy a context table:

1. In the 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 context table records

To view a list of context table records:

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 for which you want to view the context table, select **Go to context tables**.

   This opens the **Correlator context tables** window.

4. In the **Name** column, select the relevant context table.

The list of records for the selected context table is displayed.

The list contains the following data:

- **Key** is the composite key of the record. It is comprised by one or more values of key fields, separated by the "|" character. If one of the key field values is absent, the separator character is still displayed.

  For example, a record key consists of three fields: `DestinationAddress`, `DestinationPort`, and `SourceUserName`. If the last two fields do not contain values, the record key is displayed as follows: `43.65.76.98| |` .

- **Record repetitions** is the total number of times the record was mentioned in events and identical records were downloaded when importing context tables to KUMA.

- **Expiration date** – date and time when the record must be deleted.

  If the **TTL** field had the value of 0 when the context table was created, the records of this context table are retained for 36,000 days (approximately 100 years).

- **Updated** is the date and time when the context table was updated.

Searching context table records

To find a record in the context table:

1. In the 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 in whose context table you want to find a record, select **Go to context tables**.

   This opens the **Correlator context tables** window.

4. In the **Name** column, select your context table.

   This opens a window with the records of the selected context table.

5. In the **Search** field, enter the record key value or several characters from the key.

The list of context table records displays only the records whose key contains the entered characters.

If the your search query matches records with empty key values, the text <Nothing found> is displayed in the widget on the **Dashboard**. We recommend clarifying the conditions of your search query.

Adding a context table record

To add a record to the context table:

1. In the 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 add a record, select **Go to context tables**.

   This opens the **Correlator context tables** window.

4. In the **Name** column, select the relevant context table.

   The list of records for the selected context table is displayed.

5. Click **Add**.

   The **Create record** window opens.

6. In the **Value** field, specify the values for fields in the **Field** column.

   KUMA takes field names from the correlation rules with which the context table is associated. These names are not editable. The list of fields cannot be edited.

   If you do not specify some of the field values, the missing fields, including key fields, are populated with default values. The key of the record is determined from the full set of fields, and the record is added to the table. If an identical key already exists in the table, an error is displayed.

   List of default field values ?

Field type	Default value
Integer	0
Floating point number	0.0
String	""
Boolean	false
IP address	"0.0.0.0"
Timestamp	0
Integer list	[]
Float list	[]
List of strings	[]
Boolean list	[]
Timestamp list	[]
IP list	[]

7. Click the **Save** button.

The record is added.

Editing a context table record

To edit a record in the context table:

1. In the 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 for which you want to edit the context table, select **Go to context tables**.
   This opens the **Correlator context tables** window.
4. In the **Name** column, select the relevant context table.
   The list of records for the selected context table is displayed.
5. Click on the row of the record that you want to edit.
6. Specify your values in the **Value** column.
7. Click the **Save** button.

The record is overwritten.

Restrictions when editing a record:

- The value of the key field of the record is not available for editing. You can change it by exporting and importing a record.

- Field names in the **Field** column are not editable.
- The values in the **Value** column must meet the following requirements:
  - greater than or equal to 0 for fields of the **Timestamp** and **Timestamp list** types.
  - IPv4 or IPv6 format for fields of the **IP address** and **IP list** types.
  - is **true** or **false** for a **Boolean** field.

Deleting a context table record

To delete records from a context table:

1. In the 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 from whose context table you want to delete a record, select **Go to context tables**.

   This opens the **Correlator context tables** window.
4. In the **Name** column, select the relevant context table.

   The list of records for the selected context table is displayed.
5. Select the check boxes next to the records you want to delete.

   To delete all records, select the check box next to the **Key** column.

   > At least one check box must be selected.
6. Click the **Delete** button.
7. Click **OK**.

The records will be deleted.

Importing data into a context table

To import data to a context table:

1. In the 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.

Exporting data from a context table

To export data from 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 whose context table you want to export, select **Go to context tables**.

   This opens the **Correlator context tables** window.
4. Select the check box next to your context table and click **Export**.

   The context table is downloaded to your computer in JSON format. The name of the downloaded file reflects the name of the context table. The order of the fields in the file is not defined.

Example of incident investigation with KUMA

Detecting an attack in the organization IT infrastructure using KUMA 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 Administration Kit, 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 application 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 program 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 program 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 program 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 application settings. Possible date format options:

- English localization: YYYY-MM-DD.
- Russian localization: DD.MM.YYYY.

Dashboard

In the **Dashboard** section, you can monitor the security status of your organization's network.

The dashboard is a set of widgets that display network security data analytics. You can view data only for those tenants to which you have access.

A selection of widgets used in the dashboard is called a *layout*. You can create layouts manually or use predefined layouts. You can edit widget settings in predefined layouts as necessary. By default, the dashboard displays the Alerts Overview predefined layout.

Only users with the Main administrator, Tenant administrator, Tier 2 analyst, and Tier 1 analyst roles can create, edit, or delete layouts. Users accounts with all roles can view layouts and set default layouts. If a layout is set as default, that layout is displayed for the account every time the user navigates to the **Dashboard** section. The selected default layout is saved for the current user account.

The information on the dashboard is updated in accordance with the schedule configured in layout settings. If necessary, you can force the update of the data.

For convenient presentation of information on the dashboard, you can enable TV mode. This mode lets you view the dashboard in full-screen mode in FullHD resolution. In TV mode, you can also configure a slide show display for the selected layouts.

Creating a dashboard layout

To create a layout:

1. Open the KUMA web interface and select the **Dashboard** section.
2. Open the drop-down list in the top right corner of the **Dashboard** window and select **Create layout**.

   The **New layout** window opens.
3. In the **Tenants** drop-down list, select the tenants that will own the created layout and whose data will be used to fill the widgets of the layout.

   The selection of tenants in this drop-down list does not matter if you want to create a universal layout (see below).
4. In the **Time period** drop-down list, select the time period from which you require analytics:
   - **1 hour**
   - **1 day** (this value is selected by default)
   - **7 days**
   - **30 days**
   - **In period**—receive analytics for the custom time period. The time period is set using the calendar that is displayed when this option is selected.

> The upper boundary of the period is not included in the time slice defined by it. In other words, to receive analytics for a 24-hour period, you should configure the period as Day 1, 00:00:00 – Day 2, 00:00:00 instead of Day 1, 00:00:00 – Day 1, 23:59:59.

5. In the **Refresh every** drop-down list, select how often data should be updated in layout widgets:

   - **1 minute**
   - **5 minutes**
   - **15 minutes**
   - **1 hour** (this value is selected by default)
   - **24 hours**

6. In the **Add widget** drop-down list, select the required widget and configure its settings.

   You can add multiple widgets to the layout.

   You can also drag widgets around the window and resize them using the ⤡ button that appears when you hover the mouse over a widget.

   You can edit or delete widgets added to the layout by clicking the ⚙ icon and selecting **Edit** to change their configuration or **Delete** to delete them from the layout.

   - Adding widgets

     > *To add widget:*
     >
     > 1. Click the **Add widget** drop-down list and select required widget.
     >
     >   The window with widget parameters opens. You can see how the widget will look like by clicking the **Preview** button.
     >
     > 2. Configure widget parameters and click the **Add** button.

   - Editing widget

     > *To edit widget:*
     >
     > 1. Hover the mouse over the required widget and clicking the ⚙ icon that appears.
     >
     > 2. In the drop-down list select **Edit**.
     >
     >   The window with widget parameters opens. You can see how the widget will look like by clicking the **Preview** button.
     >
     > 3. Update widget parameters and click the **Save** button.

7. In the **Layout name** field, enter a unique name for this layout. Must contain 1 to 128 Unicode characters.

8. If necessary, click the ⚙ icon on the right of the layout name field and select the check boxes next to the additional layout settings:

   - **Universal**—if you select this check box, layout widgets display data from tenants that you select in the **Selected tenants** section in the menu on the left. This means that the data in the layout widgets will change based on your selected tenants without having to edit the layout settings. For universal layouts, tenants selected in the **Tenants** drop-down list are not taken into account.

If this check box is cleared, layout widgets display data from the tenants that are selected in the **Tenants** drop-down list in the layout settings. If any of the tenants selected in the layout are not available to you, their data will not be displayed in the layout widgets.

> You cannot use the Active Lists widget in universal layouts.

> Universal layouts can only be created and edited by General administrators. Such layouts can be viewed by all users.

- **Show CII-related data**—if you select this check box, layout widgets will also show data on assets, alerts, and incidents related to critical information infrastructure (CII). In this case, these layouts will be available for viewing only by users whose settings have the **Access to CII facilities** check box selected.

  If this check box is cleared, layout widgets will not display data on CII-related assets, alerts, and incidents, even if the user has access to CII objects.

9. Click **Save**.

The new layout is created and is displayed in the **Dashboard** section of the KUMA web interface.

Selecting a dashboard layout

To select a dashboard layout:

1. Expand the list in the upper right corner of the **Dashboard** window.
2. Select the relevant layout.

The selected layout is displayed in the **Dashboard** section of the KUMA web interface.

Selecting a dashboard layout as the default

To set a dashboard layout as the default:

1. In the KUMA web interface, select the **Dashboard** section.
2. Expand the list in the upper right corner of the **Dashboard** window.
3. Hover the mouse cursor over the relevant layout.
4. Click the ☆ icon.

The selected layout is displayed on the dashboard by default.

Editing a dashboard layout

To edit a dashboard layout:

1. In the KUMA web interface, select the **Dashboard** section.

2. Expand the list in the upper right corner of the window.

3. Hover the mouse cursor over the relevant layout.

4. Click the ✎ icon.

   The **Customizing layout** window opens.

5. Make the necessary changes. The settings that are available for editing are the same as the settings available when creating a layout.

6. Click the **Save** button.

The dashboard layout is edited and displayed in the **Dashboard** section of the KUMA web interface.

> If the layout is deleted or assigned to a different tenant while are making changes to it, an error is displayed when you click **Save**. The layout is not saved. Refresh the KUMA web interface page to see the list of available layouts in the drop-down list.

Deleting a dashboard layout

To delete layout:

1. In the KUMA web interface, select the **Dashboard** section.

2. Expand the list in the upper right corner of the window.

3. Hover the mouse cursor over the relevant layout.

4. Click the 🗑 icon and confirm this action.

The layout is deleted.

Enabling and disabling TV mode

> It is recommended to create a separate user with the minimum required set of right to display analytics in TV mode.

To enable TV mode:

1. In the KUMA web interface, select the **Dashboard** section.

2. Click the ⚙ button in the upper-right corner.

   The **Settings** window opens.

3. Move the **TV mode** toggle switch to the **Enabled** position.

4. To configure the slideshow display of the layouts, do the following:

   a. Move the **Slideshow** toggle switch to the **Enabled** position.

   b. In the **Timeout** field, indicate how many seconds to wait before switching layouts.

   c. In the **Queue** drop-down list, select the layouts to view. If no layout is selected, the slideshow mode displays all layouts available to the user one after another.

   d. If necessary, change the order in which the layouts are displayed using the ⠿ button to drag and drop them.

5. Click the **Save** button.

TV mode will be enabled. To return to working with the KUMA web interface, disable TV mode.

To disable TV mode:

1. Open the KUMA web interface and select the **Dashboard** section.

2. Click the ⚙ button in the upper-right corner.
   The **Settings** window opens.

3. Move the **TV mode** toggle switch to the **Disabled** position.

4. Click the **Save** button.

TV mode will be disabled. The left part of the screen shows a pane containing sections of the KUMA web interface.

When you make changes to the layouts selected for the slideshow, those changes will automatically be applied to the active slideshow sessions.

Predefined dashboard layouts

KUMA comes with a set of predefined layouts: The default refresh period for predefined layouts is **Never**. You can edit these layouts as needed.

Predefined layouts

Layout name	Description of widgets in the layout
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 layout. • Alerts distribution—number of alerts created during the period configured for the widget. • Alerts by priority—number of unclosed alerts grouped by their priority.

	• Alerts by 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 layout. • Incidents distribution—number of incidents created during the period configured for the widget. • Incidents by priority—number of unclosed incidents grouped by their priority. • 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 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 matrix techniques detected by the EDR system. • EDR. Top-10 MITRE tactic detections — visualizes the 10 most common MITRE 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.
[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.

Reports

You can configure KUMA to regularly generate reports about KUMA processes.

Reports are generated using *report templates* that are created and stored on the **Templates** tab of the **Reports** section.

Generated reports are stored on the **Generated reports** tab of the **Reports** section.

> 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 Application 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.

Report template

Report templates are used to specify the analytical data to include in the report, and to configure how often reports must be generated. 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 has **the following columns** ?:

> You can configure a set of table columns and their order, as well as change data sorting:
>
> - You can enable or disable the display of columns in the menu that can be opened by clicking the icon ⚙.
> - You can change the order of columns by dragging the column headers.
> - If a table column header is green, you can click it to sort the table based on that column's data.

- **Name**—the name of the report template.

  You can sort the table by this column by clicking the title and selecting **Ascending** or **Descending**.

  You can also search report templates by using the **Search** field that opens when you click the **Name** column title.

  > Regular expressions are used when searching for report templates.

- **Schedule**—the rate at which reports must be generated using the template. If the report schedule was not configured, the `disabled` value is displayed.
- **Created by**—the name of the user who created the report template.
- **Updated**—the date when the report template was last updated.

  You can sort the table by this column by clicking the title and selecting **Ascending** or **Descending**.
- **Last report**—the date and time when the last report was generated based on the report template.
- **Send by email**—the check mark is displayed in this column for the report templates that notify users about generated reports via email notifications.
- **Tenant**—the name of the tenant that owns the report template.

You can click the name of the report template to open the drop-down list with available commands:

- **Run report**—use this option to generate report immediately. The generated reports are displayed on the **Generated reports** tab.
- **Edit schedule**—use this command to configure the schedule for generating reports and to define users that must receive email notifications about generated reports.
- **Edit report template**—use this command to configure widgets and the time period for extracting analytics.
- **Duplicate report template**—use this command to create a copy of the existing report template.
- **Delete report template**—use this command to delete the report template.

Creating report template

To create report template:

1. Open the KUMA web interface and select **Reports** → **Templates**.

2. Click the **New template** button.

   The **New report template** window opens.

3. In the **Tenants** drop-down list, select one or more tenants that will own the layout being created.

4. In the **Time period** drop-down list, select the time period from which you require analytics:

   - **This day** (this value is selected by default)

   - **This week**

   - **This month**

   - **In period**—receive analytics for the custom time period.

     > The upper boundary of the period is not included in the time slice defined by it. In other words, to receive analytics for a 24-hour period, you should configure the period as Day 1, 00:00:00 – Day 2, 00:00:00 instead of Day 1, 00:00:00 – Day 1, 23:59:59.

   - **Custom**—receive analytics for the last N days/weeks/months/years.

5. In the **Retention** field, specify how long you want to store reports that are generated according to this template.

6. In the **Template name** field, enter a unique name for the report template. Must contain 1 to 128 Unicode characters.

7. In the **Add widget** drop-down list, select the required widget and configure its settings.

   You can add multiple widgets to the report template.

   You can also drag widgets around the window and resize them using the ⤡ button that appears when you hover the mouse over a widget.

   You can edit or delete widgets added to the layout by hovering the mouse over them, clicking the ⚙ icon that appears and selecting **Edit** to change their configuration or **Delete** to delete them from layout.

   - **Adding widgets**

     > *To add widget:*
     >
     > 1. Click the **Add widget** drop-down list and select required widget.
     >
     >   The window with widget parameters opens. You can see how the widget will look like by clicking the **Preview** button.
     >
     > 2. Configure widget parameters and click the **Add** button.

   - **Editing widget**

To edit widget:

1. Hover the mouse over the required widget and clicking the ⚙ icon that appears.
2. In the drop-down list select **Edit**.

   The window with widget parameters opens. You can see how the widget will look like by clicking the **Preview** button.
3. Update widget parameters and click the **Save** button.

8. You can change logo in the report template by clicking the **Upload logo** button.

   When you click the **Upload logo** button, the Upload window opens and lets you choose the image file for the logo. The image must be a .jpg, .png, or .gif file no larger than 3 MB.

   The added logo is displayed in the report instead of KUMA logo.
9. If necessary, select the **Show CII-related data** check box to display data on assets, alerts, and incidents related to critical information infrastructure (CII) in the layout widgets. In this case, these layouts will be available for viewing only by users whose settings have the **Access to CII facilities** check box selected.

   If this check box is cleared, layout widgets will not display data on CII-related assets, alerts, and incidents, even if the user has access to CII objects.
10. Click **Save**.

The new report template is created and is displayed on the **Reports** → **Templates** tab of the KUMA web interface. You can run this report manually. If you want to have the reports generated automatically, you must configure the schedule for that.

Configuring report schedule

To configure the report schedule:

1. Open the KUMA web interface and select **Reports** → **Templates**.
2. In the report templates table, click the name of an existing report template and select **Edit schedule** in the drop-down list.

   The **Report settings** window opens.
3. If you want the report to be generated regularly:

   a. Turn on the **Schedule** toggle switch.

   In the **Recur every** group of settings, define how often the report 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.

   b. In the **Time** field, enter the time when the report must be generated. You can enter the value manually or using the clock icon.
4. To select the report format and specify the report recipients, configure the following settings:

   a. In the **Send to** group of settings, click **Add**.

b. In the **Add emails** window that opens, in the **User group** section, click **Add group**.

c. In the field that appears, specify the email address and press **Enter** or click outside the entry field—the email address will be added. You can add more than one address. Reports are sent to the specified addresses every time you generate a report manually or KUMA generates a report automatically on schedule.

> You should configure an SMTP connection so that generated reports can be forwarded by email.

If the recipients who received the report by email are KUMA users, they can download or view the report by clicking the links in the email. 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 without attachments, the recipients will have access to reports only by links and only with authorization in KUMA, without restrictions on roles or tenants.

d. In the drop-down list, select the report format to send. Available formats: PDF, HTML, CSV, split CSV, Excel.

5. Click **Save**.

Report schedule is configured.

Editing report template

To edit report template:

1. Open the KUMA web interface and select **Reports** → **Templates**.
2. In the report templates table click the name of the report template and select **Edit report template** in the drop-down list.

   The **Edit report template** window opens.

   > You can also open this window on the **Reports** → **Generated reports** tab by clicking the name of a generated report and selecting in the drop-down list **Edit report template**.

3. Make the necessary changes:
   - Change the list of tenants that own the report template.
   - Update the time period from which you require analytics.
   - Add widgets

     To add widget:

     1. Click the **Add widget** drop-down list and select required widget.

        The window with widget parameters opens. You can see how the widget will look like by clicking the **Preview** button.

     2. Configure widget parameters and click the **Add** button.

- Change widgets positions by dragging them.
- Resize widgets using the ⤡ button that appears when you hover the mouse over a widget.
- Edit widgets

  > *To edit widget:*
  >
  > 1. Hover the mouse over the required widget and clicking the ⚙ icon that appears.
  > 2. In the drop-down list select **Edit**.
  >
  >   The window with widget parameters opens. You can see how the widget will look like by clicking the **Preview** button.
  > 3. Update widget parameters and click the **Save** button.

- Delete widgets by hovering the mouse over them, clicking the ⚙ icon that appears, and selecting **Delete**.
- In the field to the right from the **Add widget** drop-down list enter a new name of the report template. Must contain 1 to 128 Unicode characters.
- Change the report logo by uploading it using the **Upload logo** button. If the template already contains a logo, you must first delete it.
- Change how long reports generated using this template must be stored.
- If necessary, select or clear the **Show CII-related data** check box.

4. Click **Save**.

The report template is updated and is displayed on the **Reports** → **Templates** tab of the KUMA web interface.

Copying report template

To create a copy of a report template:

1. Open the KUMA web interface and select **Reports** → **Templates**.
2. In the report templates table, click the name of an existing report template, and select **Duplicate report template** in the drop-down list.

   The **New report template** window opens. The name of the widget is changed to `<Report template> - copy`.
3. Make the necessary changes:
   - Change the list of tenants that own the report template.
   - Update the time period from which you require analytics.
   - Add widgets

> *To add widget:*
>
> 1. Click the **Add widget** drop-down list and select required widget.
>
>   The window with widget parameters opens. You can see how the widget will look like by clicking the **Preview** button.
>
> 2. Configure widget parameters and click the **Add** button.

- Change widgets positions by dragging them.
- Resize widgets using the ⤡ button that appears when you hover the mouse over a widget.
- Edit widgets

> *To edit widget:*
>
> 1. Hover the mouse over the required widget and clicking the ⚙ icon that appears.
>
> 2. In the drop-down list select **Edit**.
>
>   The window with widget parameters opens. You can see how the widget will look like by clicking the **Preview** button.
>
> 3. Update widget parameters and click the **Save** button.

- Delete widgets by hovering the mouse over them, clicking the ⚙ icon that appears, and selecting **Delete**.
- In the field to the right from the **Add widget** drop-down list enter a new name of the report template. Must contain 1 to 128 Unicode characters.
- Change the report logo by uploading it using the **Upload logo** button. If the template already contains a logo, you must first delete it.

4. Click **Save**.

The report template is updated and is displayed on the **Reports** → **Templates** tab of the KUMA web interface.

Deleting report template

To delete report template:

1. Open the KUMA web interface and select **Reports** → **Templates**.
2. In the report templates table, click the name of the report template, and select **Delete report template** in the drop-down list.

   A confirmation window opens.
3. If you want to delete only the report template, click the **Delete** button.
4. 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.

Generated reports

All reports are generated using report templates. Generated reports are available on the **Generated reports** tab of the **Reports** section and are displayed in the table with **the following columns**:

> You can configure a set of table columns and their order, as well as change data sorting:
>
> - You can enable or disable the display of columns in the menu that can be opened by clicking the icon ⚙.
> - You can change the order of columns by dragging the column headers.
> - If a table column header is green, you can click it to sort the table based on that column's data.

- **Name**—the name of the report template.

  You can sort the table by this column by clicking the title and selecting **Ascending** or **Descending**.
- **Time period**—the time period for which the report analytics were extracted.
- **Last report**—date and time when the report was generated.

  You can sort the table by this column by clicking the title 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.

You can click the name of a report to open the drop-down list with available commands:

- **Open report**—use this command to open the report data window.
- **Save as**—use this command to save the generated report in the desired format. Available formats: HTML, PDF, CSV, split CSV, Excel.
- **Run report**—use this option to generate report immediately. Refresh the browser window to see the newly generated report in the table.
- **Edit report template**—use this command to configure widgets and the time period for extracting analytics.
- **Delete report**—use this command to delete the report.

Viewing reports

To open report:

1. Open the KUMA web interface and select **Reports** → **Generated reports**.
2. In the report table, click the name of the generated report, and select **Open report** in the drop-down list.

The new browser window opens with the widgets displaying report analytics. If a widget displays data on events, alerts, incidents, or active lists, you can click its header to open the corresponding section of the KUMA web interface with an active filter and/or search query that is used to display data from the widget. Widgets are subject to default restrictions.

To download the data displayed on each widget in CSV format with UTF-8 encoding, press the **CSV** button. The downloaded file name has the format <widget name>_<download date (YYYYMMDD)>_<download time (HHMMSS)>.CSV.

To view the full data, download the report in the CSV format with the specified settings from the request.

3. You can save the report in the desired format by using 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 select **Reports** → **Templates**.
2. In the report templates table, click a report template name and select **Run report** in the drop-down list.

> You can also generate report from the **Reports** → **Generated reports** tab by clicking the name of an existing report and in the drop-down list selecting **Run report**.

The report is generated and is displayed on the **Reports** → **Generated reports** tab.

To generate reports automatically, configure the report schedule.

Saving reports

To save the report in the desired format:

1. Open the KUMA web interface and select **Reports** → **Generated reports**.
2. In the report table, click the name of the generated report, and in the drop-down list select **Save as**. Then select the desired 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.

Deleting reports

To delete report:

1. Open the KUMA web interface and select **Reports** → **Generated reports**.
2. In the report table, click the name of the generated report, and in the drop-down list select **Delete report**.

   A confirmation window opens.

3. Click **OK**.

Widgets

Widgets let you monitor the operation of the application.

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 analytics related to alerts.

  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. 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.
  - **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 related to 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 analytics related to incidents.

The group includes the following widgets:

  - **Active incidents**—number of incidents that have not been closed.
  - **Unassigned incidents**—number of incidents that have the **Opened** status.
  - **Incidents distribution**—number of incidents created during the period configured for the widget.
  - **Incidents by assignee**—number of incidents with the **Assigned** status. The grouping is 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 analytics related to sources of events. The group includes the following widgets:
  - **Top event sources by alerts number**—number of unclosed alerts grouped by event source.
  - **Top event sources by convention rate**—number of events associated with unclosed alerts. The grouping is by event source.

    > In some cases, the number of alerts generated by sources may be inaccurate. To obtain accurate statistics, it is recommended to specify the Device Product event field as unique in the correlation rule, and enable storage of all base events in a correlation event. However, correlation rules with these settings consume more resources.

- Users—group for analytics related to users from processed events. The group includes the following widgets:
  - **Affected users in alerts**—number of accounts related to unclosed alerts.
  - **Number of AD users**—number of Active Directory accounts received via LDAP during the period configured for the widget.

In the events table, in the event details area, in the alert window, and in the widgets, the names of assets, accounts, and services are displayed instead of the IDs as the values of the SourceAssetID, DestinationAssetID, DeviceAssetID, SourceAccountID, DestinationAccountID, and ServiceID fields. When exporting events to a file, the IDs are saved, but columns with names are added to the file. The IDs are also displayed when you point the mouse over the names of assets, accounts, or services.

Searching for fields with IDs is only possible using IDs.

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 (4).
- Table (▦).
- Bar chart (☰).
- Date Histogram (📊).
- Line 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 (30д). You can view the start and end dates of the period and the time of the last update by hovering the mouse cursor over this icon.

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.

Basics of managing "Pie chart" graphs

A pie chart is displayed under the list of tenants. You can left-click the selected segment of the diagram to go to the relevant section of the KUMA web interface. The data in that section is sorted in accordance with the filters and/or search query specified in the widget.

Under the period icon, you can see the number of events, active lists, assets, alerts, or incidents grouped by the selected criteria for the data display period.

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:
The **Number of assets** widget displays the total number of assets added to KUMA.

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.

Basics of managing "Bar chart" graphs

A bar chart is displayed below the list of tenants. You can left-click the selected diagram section to go to the **Events** section of the KUMA web interface. The data in that section is sorted in accordance with the filters and/or search query specified in the widget. To the right of the chart, the same data is represented as a table.

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.

Basics of managing "Date Histogram" graphs

A date histogram is displayed below the list of tenants. You can left-click the selected section of the chart to go to the **Events** section of the KUMA web interface with the relevant data. The data in that section is sorted in accordance with the filters and/or search query specified in the widget. To the right of the chart, the same data is represented as a table.

> Example:
> In the **Events** widget, for which the SQL query `SELECT count(ID) AS `metric`, Timestamp AS `value` FROM `events` GROUP BY Timestamp ORDER BY `metric` DESC LIMIT 250` is specified, the x-axis of the diagram corresponds to event creation date, and the y-axis corresponds to the approximate number of events. Events are grouped by creation date in descending order.

Basics of managing "Line chart" graphs

A line chart is displayed below the list of tenants. You can left-click the selected section of the chart to go to the **Events** section of the KUMA web interface with the relevant data. The data in that section is sorted in accordance with the filters and/or search query specified in the widget. To the right of the chart, the same data is represented as a table.

> 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.

Special considerations for displaying data in widgets

Limitations for the displayed data

For improved readability, KUMA has limitations on the data displayed in widgets depending on its type:

- Pie chart displays a maximum of 20 slices.
- Bar chart displays a maximum of 40 bars.
- Table displays a maximum of 500 entries.
- Date histogram displays a maximum of 365 days.

Data that exceeds the specified limitations is displayed in the widget in the **Other** category.

You can download the full data used for building analytics in the widget in CSV format.

Summing up the data

The format of displaying the total sum of data on date histogram, bar chart and pie chart depends on the locale:

- English locale: decades (every three digits) are separated by commas, the decimal part is separated by a period.
- Russian locale: decades (every three digits) are separated by spaces, the decimal part is separated by a comma.

Creating a widget

You can create a widget in a dashboard layout while creating or editing the layout.

To create a widget:

1. Create a layout or switch to editing mode for the selected layout.
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 **Add**.

The widget appears in the dashboard layout.

Editing a widget

To edit widget:

1. In the KUMA web interface, select the **Dashboard** section.
2. Expand the list in the upper right corner of the window.
3. Hover the mouse cursor over the relevant layout.
4. Click the ✎ button.

   The **Customizing layout** window opens.
5. In the widget you want to edit, click ⚙.
6. Select **Edit**.

   This opens the widget settings window.
7. Edit the widget settings.
8. Click **Save** in the widget settings window.
9. Click Save in the **Customizing layout** window.

The widget is edited.

Deleting a widget

To delete a widget:

1. In the KUMA web interface, select the **Dashboard** section.
2. Expand the list in the upper right corner of the window.
3. Hover the mouse cursor over the relevant layout.
4. Click the ✎ button.
   The **Customizing layout** window opens.
5. In the widget you want to delete, click ⚙.
6. Select **Delete**.
7. This opens a confirmation window; in that window, click **OK**.
8. Click the **Save** button.

The widget is deleted.

Widget settings

This section describes the settings of all widgets available in KUMA.

"Events" widget

You can use the **Events** widget to get analytics based on SQL queries.

When creating this type of widget, you must set values for the following settings:

The ☰ tab:

- **Graph** is the type of the graph. The following graph types are available:
  - **Pie chart**.
  - **Bar chart**.
  - **Counter**.
  - **Line chart**.
  - **Table**.
  - **Date Histogram**.
- **Tenant** is the tenant for which data is displayed in the widget.
  You can select multiple tenants.
  By default, data is displayed for tenants that have been selected in layout settings.
- **Period** is the period for which data is displayed in the widget. The following periods are available:

  - **As layout** means data is displayed for the period selected for the layout.

    This is the default setting.

  - **1 hour**—data is displayed for the previous hour.

  - **1 day**—data is displayed for the previous day.

  - **7 days**—data is displayed for the previous 7 days.

  - **30 days**—data is displayed for the previous 30 days.

  - **In period**—data is displayed for a custom time period.

    If you select this option, use the opened calendar to select the start and end dates of the period and click **Apply Filter**. The date and time format depends on your operating system's settings. You can also manually change the date values if necessary.

    > The upper boundary of the period is not included in the time slice defined by it. In other words, to receive analytics for a 24-hour period, you should configure the period as Day 1, 00:00:00 – Day 2, 00:00:00 instead of Day 1, 00:00:00 – Day 1, 23:59:59.

- **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.

- **Storage** is the storage that is searched for events.

- The SQL query field (☰) lets you manually enter a query for filtering and searching events.

  You can also create a query in Builder by clicking ☰.

  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.

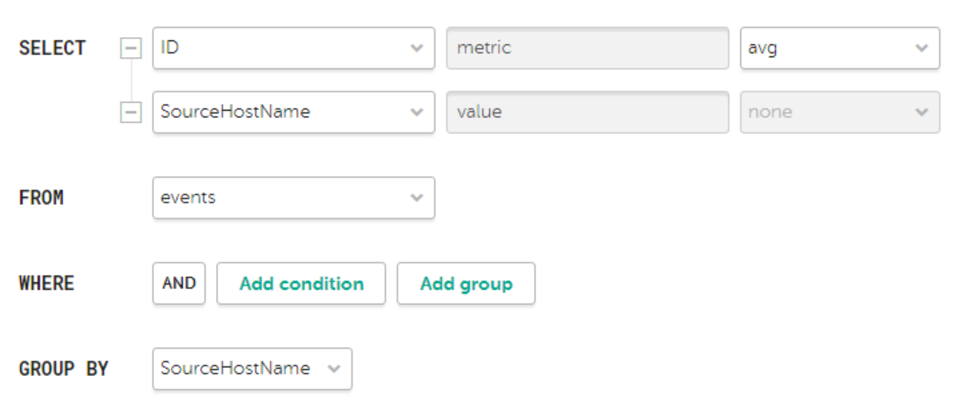
      - In the drop-down list to the left, select the value that will be used for sorting.

    - 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



SELECT ID metric avg

SourceHostName value none

FROM events

WHERE

GROUP BY SourceHostName

Search condition parameters for the widget showing average bytes received per host

The "metric" and "value" aliases in SQL queries cannot be edited for any type of event analytics widget, except tables.

Aliases in widgets 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` ``.

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 ``.

In the **Counter** type widgets you must specify the method of data processing for the values of the `SELECT` function: `count`, `max`, `min`, `avg`, `sum`.

The ⚙ tab:

The tab is displayed if on the ☰ tab in the **Graph** field you have selected one of the following values: **Bar chart**, **Line chart**, **Date Histogram**.

- The **Y-min** and **Y-max** values set the scale of the Y axis.
- The **X-min** and **X-max** values set the 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**.

- **Line-width** is the width of the line on the graph. This field is displayed for the "Line chart" graph type.
- **Point size** is the size of the pointer on the graph. This field is displayed for the "Line chart" graph type.

The 🔧 tab:

- **Name** is the name of the widget.
- **Description** is the description of the widget.
- **Color** is a drop-down list where you can select the color for displaying information:
  - **default** for your browser's default font color
  - **green**
  - **red**
  - **blue**
  - **yellow**
- **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 total** shows sums total of the values.
- **Legend** displays a legend for 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 off by default.
- **Decimals**—the field to enter the number of decimals to which the displayed value must be rounded off.
- **Period segments length** (available for graphs of the **Date Histogram** type) sets the length of segments into which you want to divide the period.

"Active lists" widget

You can use the **Active lists** widget to get analytics based on SQL queries.

When creating this type of widget, you must set values for the following settings:

The ⊟ tab:

- **Graph** is the type of the graph. The following graph types are available:
  - **Bar chart**.
  - **Pie chart**.
  - **Counter**.
  - **Table**.
- **Tenant** is the tenant for which data is displayed in the widget.

  You can select multiple tenants.

  By default, data is displayed for tenants that have been selected in layout settings.
- **Correlator** is the name of the correlator that contains the active list for which you want to receive data.
- **Active list** is 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.

- The SQL query 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 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.

- **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.

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.

> Sample SQL queries for receiving analytics based on active lists:
>
> - SELECT * FROM `records` WHERE "Event source" = 'Johannesburg' LIMIT 250
>   This query returns the key of the active list where the field name is "Event source" and the value of this field is "Johannesburg".
>
> - SELECT count(_key) AS metric, Status AS value FROM `records` GROUP BY value ORDER BY metric DESC LIMIT 250
>   Query for a pie chart, which returns the number of keys in the active list ('count' aggregation over the '_key' field) and all variants of the Status custom field. The widget displays a pie chart with the total number of records in the active list, divided proportionally by the number of possible values for the Status field.
>
> - SELECT Name, Status, _count AS Number FROM `records` WHERE Description ILIKE '%ftp%' ORDER BY Name DESC LIMIT 250
>   Query for a table, which returns the values of the Name and Status custom fields, as well as the service field '_count' for those records of the active list in which the value of the Description custom field matches ILIKE '%ftp%'. The widget displays a table with the Status, Name, and Number columns.

The ⌘ tab:

> This tab is displayed if on the ☰ tab, in the **Graph** field, you have selected **Bar chart**.

- The **Y-min** and **Y-max** values set the scale of the Y axis.
- The **X-min** and **X-max** values set the 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**.

The 🔧 tab:

- **Name** is the name of the widget.
- **Description** is the description of the widget.
- **Color** is a drop-down list where you can select the color for displaying information:
  - **default** for your browser's default font color
  - **green**
  - **red**
  - **blue**
  - **yellow**

- **Horizontal** makes the histogram horizontal instead of vertical.

  When this setting is enabled, all available information is fitted into the configured widget size. If the amount of data is great, you can increase the size of the widget to display it optimally.

- **Show total** shows sums total of the values.

- **Legend** displays a legend for 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 off by default.

"Context tables" widget

You can use the **Context tables** widget to get analytics based on SQL queries.

When creating this type of widget, you must set values for the following settings:

The ☰ tab:

- **Graph** is the type of the graph. The following graph types are available:

  - **Bar chart**.

  - **Pie chart**.

  - **Counter**.

  - **Table**.

- **Tenant** is the tenant for which data is displayed in the widget.

  You can select multiple tenants.

  By default, data is displayed for tenants that have been selected in layout settings.

- **Correlator** is the name of the correlator that contains the context table for which you want to receive information.

- **Context table** is 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.

- The **SQL query field** lets you manually enter a query for filtering and searching context table data. By default, for each widget type, the field contains a query that obtains the context table schema and the key by key fields.

  The query structure is similar to that used in event search.

  When creating a query based on context tables, you must consider the following:

  - For the FROM function, you must specify the `records` value.

- You can get data only for the fields specified in the context table schema.
- You can use supported features of ClickHouse.
- If you want to receive data for fields whose names contain spaces and Cyrillic characters, you must also enclose such names in quotes in the query:
  - In the SELECT function, enclose aliases in double quotes or backticks: "alias", `another alias`.
  - In the ORDER BY function, enclose aliases in backticks: `another alias`.
  - Event field values are enclosed in straight quotes: WHERE DeviceProduct = 'Microsoft'.

  Names of event fields do not need to be enclosed in quotes.

  > If the name of an active list field begins or ends with spaces, these spaces are not displayed by the widget. The field name must not contain spaces only.

  If the values of the active list fields contain trailing or leading spaces, it is recommended to use the LIKE '%field value%' function to search by them.
- You can use the _count service field (how many times this record has been added to the context table), as well as custom fields.
- The "metric" and "value" aliases in SQL queries cannot be edited for any type of active lists analytics widget, except tables.
- If a date and time conversion function is used in an SQL query (for example, fromUnixTimestamp64Milli) and the field being processed does not contain a date and time, an error will be displayed in the widget. To avoid this, use functions that can handle a null value. Example: SELECT _key, fromUnixTimestamp64Milli(toInt64OrNull(DateTime)) as Date FROM `records` LIMIT 250.
- Large values for the LIMIT function may lead to browser errors.
- If you select **Counter** as the graph type, you must specify the method of data processing for the values of the SELECT function: count, max, min, avg, sum.
- **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.

Special considerations when using aliases in SQL functions and SELECT statements: you may use double quotes and backquotes: ",`.
When using spaces or Cyrillic characters, the alias must be enclosed in double quotes: "Alias with a space", values must be enclosed in straight single quotes: 'Value with a space'.
When displaying data for the previous period, sorting by the count(ID) parameter may not work correctly. It is recommended to sort by the metric parameter. For example, SELECT count(ID) AS "metric", Name AS "value" FROM `events` GROUP BY Name ORDER BY metric ASC LIMIT 250.

Sample SQL queries for receiving analytics based on active lists:

- SELECT * FROM `records` WHERE "Event source" = 'Johannesburg' LIMIT 250
  This query returns the key of the active list where the field name is "Event source" and the value of this field is "Johannesburg".

- SELECT count(_key) AS metric, Status AS value FROM `records` GROUP BY value ORDER BY metric DESC LIMIT 250
  Query for a pie chart, which returns the number of keys in the active list ('count' aggregation over the '_key' field) and all variants of the Status custom field. The widget displays a pie chart with the total number of records in the active list, divided proportionally by the number of possible values for the Status field.

- SELECT Name, Status, _count AS Number FROM `records` WHERE Description ILIKE '%ftp%' ORDER BY Name DESC LIMIT 250

Query for a table, which returns the values of the Name and Status custom fields, as well as the service field '_count' for those records of the active list in which the value of the Description custom field matches ILIKE '%ftp%'. The widget displays a table with the Status, Name, and Number columns.

The tab:

This tab is displayed if on the tab, in the **Graph** field, you have selected **Bar chart**.

- The **Y-min** and **Y-max** values set the scale of the Y axis.
- The **X-min** and **X-max** values set the 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**.

The tab:

- **Name** is the name of the widget.
- **Description** is the description of the widget.
- **Color** is a drop-down list where you can select the color for displaying information:
  - **default** for your browser's default font color
  - **green**
  - **red**
  - **blue**
  - **yellow**
- **Horizontal** makes the histogram horizontal instead of vertical.

  When this setting is enabled, all available information is fitted into the configured widget size. If the amount of data is great, you can increase the size of the widget to display it optimally.
- **Show total** shows sums total of the values.
- **Legend** displays a legend for 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 off by default.

Other widgets

This section describes the settings of all widgets except the Events widgets and Active lists 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.

Settings for pie charts

- **Name** is the name of the widget.
- **Description** is the description of the widget.
- **Tenant** is the tenant for which data is displayed in the widget.

  You can select multiple tenants.

  By default, data is displayed for tenants that have been selected in layout settings.
- **Period** is the period for which data is displayed in the widget. The following periods are available:
  - **As layout** means data is displayed for the period selected for the layout.

    This is the default setting.
  - **1 hour**—data is displayed for the previous hour.
  - **1 day**—data is displayed for the previous day.
  - **7 days**—data is displayed for the previous 7 days.
  - **30 days**—data is displayed for the previous 30 days.
  - **In period**—data is displayed for a custom time period.

    If you select this option, use the opened calendar to select the start and end dates of the period and click **Apply Filter**. The date and time format depends on your operating system's settings. You can also manually change the date values if necessary.

    > The upper boundary of the period is not included in the time slice defined by it. In other words, to receive analytics for a 24-hour period, you should configure the period as Day 1, 00:00:00 – Day 2, 00:00:00 instead of Day 1, 00:00:00 – Day 1, 23:59:59.
- **Show total** shows sums total of the values.
- **Legend** displays a legend for 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 off by default.

- **Decimals**—the field to enter the number of decimals to which the displayed value must be rounded off.

Settings for counters

- **Name** is the name of the widget.

- **Description** is the description of the widget.

- **Tenant** is the tenant for which data is displayed in the widget.

  You can select multiple tenants.

  By default, data is displayed for tenants that have been selected in layout settings.

- **Period** is the period for which data is displayed in the widget. The following periods are available:

  - **As layout** means data is displayed for the period selected for the layout.

    This is the default setting.

  - **1 hour**—data is displayed for the previous hour.

  - **1 day**—data is displayed for the previous day.

  - **7 days**—data is displayed for the previous 7 days.

  - **30 days**—data is displayed for the previous 30 days.

  - **In period**—data is displayed for a custom time period.

    If you select this option, use the opened calendar to select the start and end dates of the period and click **Apply Filter**. The date and time format depends on your operating system's settings. You can also manually change the date values if necessary.

    > The upper boundary of the period is not included in the time slice defined by it. In other words, to receive analytics for a 24-hour period, you should configure the period as Day 1, 00:00:00 – Day 2, 00:00:00 instead of Day 1, 00:00:00 – Day 1, 23:59:59.

Settings for tables

- **Name** is the name of the widget.

- **Description** is the description of the widget.

- **Tenant** is the tenant for which data is displayed in the widget.

  You can select multiple tenants.

  By default, data is displayed for tenants that have been selected in layout settings.

- **Period** is the period for which data is displayed in the widget. The following periods are available:

  - **As layout** means data is displayed for the period selected for the layout.

    This is the default setting.

    - **1 hour**—data is displayed for the previous hour.
    - **1 day**—data is displayed for the previous day.
    - **7 days**—data is displayed for the previous 7 days.
    - **30 days**—data is displayed for the previous 30 days.
    - **In period**—data is displayed for a custom time period.

      If you select this option, use the opened calendar to select the start and end dates of the period and click **Apply Filter**. The date and time format depends on your operating system's settings. You can also manually change the date values if necessary.

      > The upper boundary of the period is not included in the time slice defined by it. In other words, to receive analytics for a 24-hour period, you should configure the period as Day 1, 00:00:00 – Day 2, 00:00:00 instead of Day 1, 00:00:00 – Day 1, 23:59:59.

- **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.
- **Color** is a drop-down list where you can select the color for displaying information:
    - **default** for your browser's default font color
    - **green**
    - **red**
    - **blue**
    - **yellow**
- **Decimals**—the field to enter the number of decimals to which the displayed value must be rounded off.

Settings for Bar charts and Date Histograms

The ⌘ tab:

- The **Y-min** and **Y-max** values set the scale of the Y axis.
- The **X-min** and **X-max** values set the 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**—the field to enter the number of decimals to which the displayed value must be rounded off.

The 🔧 tab:

- **Name** is the name of the widget.

- **Description** is the description of the widget.
- **Tenant** is the tenant for which data is displayed in the widget.

  You can select multiple tenants.

  By default, data is displayed for tenants that have been selected in layout settings.
- **Period** is the period for which data is displayed in the widget. The following periods are available:
  - **As layout** means data is displayed for the period selected for the layout.

    This is the default setting.
  - **1 hour**—data is displayed for the previous hour.
  - **1 day**—data is displayed for the previous day.
  - **7 days**—data is displayed for the previous 7 days.
  - **30 days**—data is displayed for the previous 30 days.
  - **In period**—data is displayed for a custom time period.

    If you select this option, use the opened calendar to select the start and end dates of the period and click **Apply Filter**. The date and time format depends on your operating system's settings. You can also manually change the date values if necessary.

    > The upper boundary of the period is not included in the time slice defined by it. In other words, to receive analytics for a 24-hour period, you should configure the period as Day 1, 00:00:00 – Day 2, 00:00:00 instead of Day 1, 00:00:00 – Day 1, 23:59:59.
- **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.
- **Color** is a drop-down list where you can select the color for displaying information:
  - **default** for your browser's default font color
  - **green**
  - **red**
  - **blue**
  - **yellow**
- **Horizontal** makes the histogram horizontal instead of vertical.

  When this setting is enabled, all available information is fitted into the configured widget size. If the amount of data is great, you can increase the size of the widget to display it optimally.
- **Show total** shows sums total of the values.
- **Legend** displays a legend for 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 off by default.

- **Period segments length** (available for graphs of the **Date Histogram** type) sets the length of segments into which you want to divide the period.

Displaying tenant names in "Active list" type widgets

If you want the names of tenants to be displayed in 'Active list' type widgets instead of tenant IDs, in correlation rules of the correlator, configure the function for populating the active list with information about the corresponding tenant.

The configuration process involves the following steps:

1. Export the list of tenants.
2. Create a dictionary of the **Table** type.
3. Import the list of tenants obtained at step 1 into the dictionary created at step 2 of these instructions.
4. Add a local variable with the **dict** function for mapping the tenant name to tenant ID to the correlation rule.

   Example:

   - Variable: `TenantName`
   - Value: `dict ('<Name of the previously created dictionary with tenants>', TenantID)`
5. Add a **Set** action to the correlation rule, which writes the value of the previously created variable to the active list in the <key>-<value> format. As the key, specify the field of the active list (for example, `Tenant`), and in the value field, specify the variable (for example, `$TenantName`).

When this rule triggers, the name of the tenant mapped by the **dict** function to the ID in the tenant dictionary is placed in the active list. When creating widgets based on active lists, the widget displays the name of the tenant instead of the tenant ID.

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 program. 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 application 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**).

Alert overflow

Each alert and its related events cannot exceed the size of 16 MB. When this limit is reached:

- New events can no longer be linked to the alert.
- The alert has an **Overflowed** tag displayed in the **Detected** column. The same tag is displayed in the **Details on alert** section of the alert details window.

Overflowed alerts should be handled as soon as possible because new events are not added to overflowed alerts. You can filter out all events that could be linked to an alert after the overflow by clicking the **All possible related events** link.

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:

- Priority (≣)—shows the importance of a possible security threat: Critical 🟥, High 🟥, Medium 🟨, or Low ⬜.
- **Name**—alert name.

> If **Overflowed** tag is displayed next to the alert name, it means the alert size has reached or is about to reach the limit and should be processed as soon as possible.

- **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.
- **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.

Click the ⚙ button to configure the displayed columns of the alerts table.

In the **Search** field, you can enter a regular expression for searching alerts based on their related assets, users, tenants, and 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 program 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 priority**—the highest priority 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.
- **Overflowed** is a tag meaning that the alert size has reached or will soon reach the limit of 16 MB and the alert must be handled. New events are not added to the overflowed alerts, but you can click the **All possible related events** link to filter all events that could be related to the alert if there were no overflow.

  A quick alert overflow may mean that the corresponding correlation rule is configured incorrectly, and this leads to frequent triggers. Overflowed alerts should be handled as soon as possible to correct the correlation rule if necessary.

Related events section

This section contains a table of events related to the alert. If you click the › 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.

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 names

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.

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 **Priority** 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 = HQ Team, OU = Groups, OU = ExchangeObjects, DC = avp, DC = ru.
  > 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 = HQ Team, OU = Groups, OU = ExchangeObjects, DC = avp, DC = ru.
  > 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.

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 period for alerts and incidents

Alerts and incidents are stored in KUMA for a year by default. This period can be changed by editing the application startup parameters in the file /usr/lib/systemd/system/kuma-core.service on the KUMA Core server.

To change the retention period for alerts and incidents:

1. Log in to the OS of the server where the KUMA Core is installed.
2. In the /usr/lib/systemd/system/kuma-core.service file, edit the following string by inserting the necessary number of days:

   ```
   ExecStart=/opt/kaspersky/kuma/kuma core --alerts.retention <number of days to store alerts and incidents> --external :7220 --internal :7210 --mongo mongodb://localhost:27017
   ```

3. Restart KUMA by running the following commands in sequence:

   a. `systemctl daemon-reload`

   b. `systemctl restart kuma-core`

   The retention period for alerts and incidents will be changed.

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.

To configure delivery of alert generation notifications based on a custom template:

1. In the KUMA web interface, open **Settings** → **Alerts** → **Notification rules**.
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 has no notification rules, click **Add tenant** and select the relevant tenant from the **Tenant** drop-down list.
3. In the **Notification rules** settings block, click **Add** and specify the notification rule settings:
   - **Name** (required)—specify the notification rule name in this field.
   - **Recipient emails** (required)—in this settings block, you can use the **Email** button to add the email addresses to which you need to send notifications about alert generation. Addresses are added one at a time.

     > Cyrillic domains are not supported. For example, a notification cannot be sent to `login@domain.us`.

   - **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 an email 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.

4. 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.

To disable notification rules for a tenant:

1. In the KUMA web interface, open **Settings** → **Alerts** → **Notification rules** 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 application 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 ⚙ icon in the top 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**.
   - **Priority**.
   - **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.
- **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 required, you can set an arbitrary period by using the calendar that opens when you select **Before date**, **After date**, or **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.
- **Priority** 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, and 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.

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 program 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.

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.

- **Priority**—the severity of the threat posed by the incident. Possible values:

  - **Critical**

  - **High**

  - **Medium**

  - **Low**

  Priority 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 composition of the fields in the tables can be changed by clicking the ⚙ 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 **Priority** 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

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 on 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 = HQ Team, OU = Groups, OU = ExchangeObjects, DC = avp, DC = ru.
        > 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 = HQ Team, OU = Groups, OU = ExchangeObjects, DC = avp, DC = ru.
        > 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:

      - **Approved**. This means the appropriate measures were taken to eliminate the security threat.

- **Not approved**. 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 program searches for incidents falling into a specified time interval that includes assets or users from the alert. In addition, the program checks whether the generated alert pertains to the tenants specified in the incidents' **Available tenants** parameter. If a matching incident is found, the program 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** parameter blocks 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	Involvement of a controlled resource in malicious software infrastructure
	Slowed operation of the resource due to a DDoS attack
	Malware infection
	Network traffic interception
	Use of a controlled resource for phishing
	Compromised user account
	Unauthorized data modification
	Unauthorized disclosure of information
	Publication of illegal information on the resource
	Distribution of spam messages from the controlled resource
	Successful exploitation of a vulnerability
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 exported incidents settings.

> 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 application 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.

   You can attach a file to the incidents with this status.

   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.

Special consideration for successful export from the KUMA hierarchical structure to NCIRCC

If multiple KUMA nodes combined into a hierarchical structure are deployed in your organization, you can forward incidents, which are received from the child KUMA nodes, from the KUMA parent nodes to NCIRCC. For this purpose, the following conditions must be met:

- Integration with NCIRCC is configured in the parent and child KUMA nodes. The **URL** and **Token** settings in the **Settings** → **NCIRCC** section are required for the parent node but are not required for the child node.
- NCIRCC integration is enabled in both nodes.

In this case, interaction with NCIRCC is performed only at the level of the node exporting the incident to NCIRCC.

Settings of the incident received from a child KUMA node cannot be changed from a parent KUMA node. If there is not enough data for performing NCIRCC export, the incident must be changed at the child KUMA node, and then exported to NCIRCC from the parent KUMA node.

Exporting data to NCIRCC

> It is impossible to export incidents that are closed in KUMA to NCIRCC if the **Description** field was not filled in at the time of closing.

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.
4. Specify the settings on the **Basic** tab of the **Export to NCIRCC** window:
   - **Category** and **Type**—specify the type and category of the incident. Only incidents of specific categories and types can be exported to NCIRCC.
   - **TLP** (required)—assign a Traffic Light Protocol marker to an incident to define the nature of information about the incident. The default value is **RED**. Available values:
     - **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.
   - **Affected system name** (required)—specify the name of the information resource where the incident occurred. You can enter up to 500,000 characters in the field.
   - **Affected system category** (required)—specify the critical information infrastructure (CII) category of your organization. If your organization does not have a CII category, select **Information resource is not a CII object**.
   - **Affected system function** (required)—specify the scope of activity of your organization. The value specified in NCIRCC integration settings is used by default.
   - **Location** (required)—select the location of your organization from the drop-down list.

- **Affected system has Internet connection**—select this check box if the assets related to this incident have an Internet connection. By default, this check box is cleared.

  If this check box is selected, the **Technical details** tab is available. This tab displays information about the assets related to the incident. See below for more details.

- **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.

5. 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.

6. If assets are attached to the incident, you can specify their settings on the **Technical details** tab.

   This tab is 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.

7. Click **Export**.

8. 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.

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 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

If an incident is assigned the **More information required** status in NCIRCC, you can attach a file to it. 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. The incident must have the **More information required** status in NCIRCC.
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.

Sending incidents involving personal information leaks to NCIRCC

KUMA 2.1.x does not have a separate section with incident parameters for submitting information about personal information leaks to NCIRCC. Since such incidents do occur and a need exists to submit information to NCIRCC, use the following solution.

To submit incidents involving personal information leaks:

1. In the KUMA web interface, in the **Incidents** section, when creating an incident involving a personal information leak, in the **Category** field, select **Notification about a computer incident**.
2. In the **Type** field, select one of the options that involves submission of information about personal information leak:
   - **Malware infection**
   - **Compromised user account**
   - **Unauthorized disclosure of information**
   - **Successful exploitation of a vulnerability**
   - **Event is not related to a computer attack**
3. In the **Description** field, enter "The incident involves a leak of personal information. Please set the status to "More information required"".
4. Click **Save**.
5. Export the incident to NCIRCC.

After NCIRCC employees set the status to "More information required" and return the incident for further editing, in your NCIRCC account, you can provide additional information about the incident in the "Details of the personal information leak" section.

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	Involvement of a controlled resource in malicious software infrastructure
	Slowed operation of the resource due to a DDoS attack
	Malware infection
	Network traffic interception
	Use of a controlled resource for phishing
	Compromised user account
	Unauthorized data modification
	Unauthorized disclosure of information
	Publication of illegal information on the resource
	Distribution of spam messages from the controlled resource
	Successful exploitation of a vulnerability
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. **Retroscan** lets you apply correlation rules to historical events if you want to debug correlation rules or analyze historical data.

To test a rule, you do not need to replay the incident in real time, instead you can run the rule in **Retroscan** mode to process historical events which include the incident of interest.

You can use a search query to define a list of historical 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.

Retroscanned events are not enriched with data from CyberTrace or the Kaspersky Threat Intelligence Portal.

Active lists are updated during retroscanning.

> A retroscan cannot be performed on selections of events obtained using SQL queries that group data and contain arithmetic expressions.

To use Retroscan:

1. In the **Events** section of the KUMA web interface, create the required event selection:
   - Select the storage.
   - Configure search expression using the constructor or search query.
   - Select the required period.
2. Open the ⋯ drop-down list and choose **Retroscan**.

The Retroscan window opens.

3. In the **Correlator** drop-down list, select the Correlator to feed selected events to.
4. In the **Correlation rules** drop-down list, select the Correlation rules that must be used when processing events. 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 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 select **Go to Events** from the drop-down list.

This opens a new browser tab containing a table of 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 for confirmation before your browser can open the new tab containing the retroscan results. For more details, please refer to the documentation for your specific browser.

Contacting Technical Support

If you are unable to find a solution to your issue in the program documentation, please contact Kaspersky Technical Support.

Kaspersky provides technical support for this program throughout its lifecycle (please refer to the product support lifecycle page).

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. Two versions are supported:

- REST API v1 — the FQDN array is not used in requests.
- REST API v2 — the FQDN array is used in requests.

REST API requests must be sent to the following address:

https://<KUMA Core FQDN>/api/<API version>/<request>

> Example:
> https://kuma.example.com:7223/api/v1
> https://kuma.example.com:7223/api/v2

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 --mongo
   mongodb://localhost:27017 --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 the specific operations that can be performed on behalf of each user. 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 that opens 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 v1 operations

Description of available requests and responses.

Viewing a list of active lists on the correlator

GET /api/v1/activeLists

The target correlator must be running.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Query parameters

Name	Data type	Mandatory	Description	Value example
correlatorID	string	Yes	Correlator service ID	00000000-0000-0000-0000-000000000000

Response

HTTP code: 200

Format: JSON

```
type Response []ActiveListInfo

type ActiveListInfo struct {
    ID      string `json:"id"`
    Name    string `json:"name"`
    Dir     string `json:"dir"`
    Records uint64 `json:"records"`
    WALSize uint64 `json:"walSize"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Correlator service ID is not specified	query parameter required	correlatorID
403	The user does not have the required role in the correlator tenant	access denied	
404	The service with the specified identifier (correlatorID) was not found	service not found	
406	The service with the specified ID (correlatorID) is not a correlator	service is not correlator	
406	The correlator did not execute the first start	service not paired	
406	The correlator tenant is disabled	tenant disabled	
50x	Failed to access the correlator API	correlator API request failed	variable
500	Failed to decode the response body received from the correlator	correlator response decode failed	variable
500	Any other internal errors	variable	variable

Import entries to an active list

POST /api/v1/activeLists/import

The target correlator must be running.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Query parameters

Name	Data type	Mandatory	Description	Value example
correlatorID	string	Yes	Correlator service ID	00000000- 0000-0000-

				0000-000000000000
activeListID	string	If activeListName is not specified	Active list ID	00000000-0000-0000-0000-000000000000
activeListName	string	If activeListID is not specified	Active list name	Attackers
format	string	Yes	Format of imported entries	csv, tsv, internal
keyField	string	For the CSV and TSV formats only	The name of the field in the header of the CSV or TSV file that will be used as the key field of the active list record. The values of this field must be unique	ip
clear	bool	No	Clear the active list before importing. If the parameter is present in the URL query, then its value is assumed to be true. The values specified by the user are ignored. Example: /api/v1/activeLists/import?clear	

Request body

Format	Contents
csv	The first line is the header, which lists the comma-separated fields. The rest of the lines are the values corresponding to the comma-separated fields in the header. The number of fields in each line must be the same.
tsv	The first line is the header, which lists the TAB-separated fields. The remaining lines are the values corresponding to the TAB-separated fields in the header. The number of fields in each line must be the same.
internal	Each line contains one individual JSON object. Data in the internal format can be received by exporting the contents of the active list from the correlator in the KUMA web console.

Response

HTTP code: 204

Possible errors

HTTP code	Description	message field value	details field value
400	Correlator service ID is not specified	query parameter required	correlatorID
400	Neither the activeListID parameter nor the activeListName parameter is specified	one of query parameters required	activeListID, activeListName
400	The format parameter is not specified	query parameter required	format
400	The format parameter is invalid	invalid query	format

		parameter value	
400	The keyField parameter is not specified	query parameter required	keyField
400	The request body has a zero-length	request body required	
400	The CSV or TSV file does not contain the field specified in the keyField parameter	correlator API request failed	line 1: header does not contain column <name>
400	Request body parsing error	correlator API request failed	line <number>: <message>
403	The user does not have the required role in the correlator tenant	access denied	
404	The service with the specified identifier (correlatorID) was not found	service not found	
404	No active list was found	active list not found	
406	The service with the specified ID (correlatorID) is not a correlator	service is not correlator	
406	The correlator did not execute the first start	service not paired	
406	The correlator tenant is disabled	tenant disabled	
406	A name search was conducted for the active list (activeListName), and more than one active list was found	more than one matching active lists found	
50x	Failed to access the correlator API	correlator API request failed	variable
500	Failed to decode the response body received from the correlator	correlator response decode failed	variable
500	Any other internal errors	variable	variable

Searching alerts

GET /api/v1/alerts

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, Interaction with NCIRCC, Access to CII.

Query parameters

Name	Data type	Mandatory	Description	Value example
page	number	No	Page number. Starts with 1. The page size is 250 entries. If the parameter is not specified, the default value is 1.	1
id	string	No	Alert ID. If the parameter is specified	00000000-0000-

			several times, then a list is generated and the logical OR operator is applied.	0000-0000-000000000000
TenantID	string	No	Alert tenant ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied. If the user does not have the required role in the specified tenant, then this tenant is ignored.	00000000-0000-0000-0000-000000000000
name	string	No	Alert name. Case-insensitive regular expression (PCRE).	alert ^My alert$
timestampField	string	No	The name of the alert field that is used to perform sorting (DESC) and search by period (from – to). lastSeen by default.	lastSeen, firstSeen
from	string	No	Lower bound of the period in RFC3339 format. <timestampField> >= <from>	2021-09-06T00:00:00Z (UTC) 2021-09-06T00:00:00.000Z (UTC, including milliseconds) 2021-09-06T00:00:00Z+00:00 (MSK)
to	string	No	Upper bound of the period in RFC3339 format. <timestampField> <= <to>	2021-09-06T00:00:00Z (UTC) 2021-09-06T00:00:00.000Z (UTC, including milliseconds) 2021-09-06T00:00:00Z+00:00 (MSK)
status	string	No	Alert status. If the parameter is specified several times, then a list is generated and the logical OR operator is applied.	new, assigned, escalated, closed
withEvents	bool	No	Include normalized KUMA events associated with found alerts in the response. If the parameter is present in the URL query, then its value is assumed to be true. The values specified by the user are ignored. Example: /api/v1/alerts?withEvents	
withAffected	bool	No	Include information about the assets and accounts associated with the found alerts in the report. If the parameter is present in the URL query, then its value is assumed to be true. The values specified by the user are ignored. Example: /api/v1/alerts?withAffected	

Response

HTTP code: 200

Format: JSON

```
type Response []Alert

type Alert struct {
    ID                string              `json:"id"`
    TenantID          string              `json:"tenantID"`
    TenantName        string              `json:"tenantName"`
    Name              string              `json:"name"`
    CorrelationRuleID string              `json:"correlationRuleID"`
    Priority          string              `json:"priority"`
    Status            string              `json:"status"`
    FirstSeen         string              `json:"firstSeen"`
    LastSeen          string              `json:"lastSeen"`
    Assignee          string              `json:"assignee"`
    ClosingReason     string              `json:"closingReason"`
    Overflow          bool                `json:"overflow"`
    Events            []NormalizedEvent `json:"events"`
    AffectedAssets    []AffectedAsset   `json:"affectedAssets"`
    AffectedAccounts  []AffectedAccount `json:"affectedAccounts"`
}

type NormalizedEvent map[string]interface{}

type AffectedAsset struct {
    ID               string          `json:"id"`
    TenantID         string          `json:"tenantID"`
    TenantName       string          `json:"tenantName"`
    Name             string          `json:"name"`
    FQDN             string          `json:"fqdn"`
    IPAddresses      []string        `json:"ipAddresses"`
    MACAddresses     []string        `json:"macAddresses"`
    Owner            string          `json:"owner"`
    OS               *OS             `json:"os"`
    Software         []Software      `json:"software"`
    Vulnerabilities  []Vulnerability `json:"vulnerabilities"`
    KSC              *KSCFields      `json:"ksc"`
    Created          string          `json:"created"`
    Updated          string          `json:"updated"`
}

type OS struct {
    Name    string `json:"name"`
    Version uint64 `json:"version"`
}

type Software struct {
    Name    string `json:"name"`
    Version string `json:"version"`
    Vendor  string `json:"vendor"`
}

type Vulnerability struct {
    KasperskyID           string   `json:"kasperskyID"`
    ProductName           string   `json:"productName"`
    DescriptionURL        string   `json:"descriptionURL"`
    RecommendedMajorPatch string   `json:"recommendedMajorPatch"`
    RecommendedMinorPatch string   `json:"recommendedMinorPatch"`
    SeverityStr           string   `json:"severityStr"`
    Severity              uint64   `json:"severity"`
    CVE                   []string `json:"cve"`
    ExploitExists         bool     `json:"exploitExists"`
    MalwareExists         bool     `json:"malwareExists"`
}

type AffectedAccount struct {
    Name             string `json:"displayName"`
    CN               string `json:"cn"`
    DN               string `json:"dn"`
    UPN              string `json:"upn"`
    SAMAccountName   string `json:"sAMAccountName"`
    Company          string `json:"company"`
    Department       string `json:"department"`
    Created          string `json:"created"`
    Updated          string `json:"updated"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Invalid value of the "page" parameter	invalid query parameter value	page
400	Invalid value of the "status" parameter	invalid status	<status>
400	Invalid value of the "timestampField" parameter	invalid timestamp field	
400	Invalid value of the "from" parameter	cannot parse from	variable
400	Invalid value of the "to" parameter	cannot parse to	variable
400	The value of the "from" parameter is greater than the value of the "to" parameter	from cannot be greater than to	
500	Any other internal errors	variable	variable

Closing alerts

POST /api/v1/alerts/close

The target correlator must be running.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, Interaction with NCIRCC, Access to CII.

Request body

Format: JSON

Name	Data type	Mandatory	Description	Value example
id	string	Yes	Alert ID	00000000-0000-0000-0000-000000000000
reason	string	Yes	Reason for closing the alert	responded, incorrect data, incorrect correlation rule

Response

HTTP code: 204

Possible errors

HTTP code	Description	message field value	details field value
400	Alert ID is not specified	id required	
400	The reason for closing the alert is not specified	reason required	
400	Invalid value of the "reason" parameter	invalid reason	
403	The user does not have the required role in the alert tenant	access denied	
404	Alert not found	alert not found	
406	Alert tenant disabled	tenant disabled	
406	Alert already closed	alert already closed	
500	Any other internal errors	variable	variable

Searching assets

GET /api/v1/assets

Information about the software of assets from KSC is not stored in KUMA and is not shown in the response.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, Access to NCIRCC, Access to CII.

Query parameters

Name	Data type	Mandatory	Description	Value example
page	number	No	Page number. Starts with 1. The page size is 250	1

			entries. If the parameter is not specified, the default value is 1.	
id	string	No	Asset ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied.	00000000-0000-0000-0000-000000000000
TenantID	string	No	Asset tenant ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied. If the user does not have the required role in the specified tenant, then this tenant is ignored.	00000000-0000-0000-0000-000000000000
name	string	No	Asset name. Case-insensitive regular expression (PCRE).	asset ^My asset$
fqdn	string	No	Asset FQDN. Case-insensitive regular expression (PCRE).	^com$ example.com
ip	string	No	Asset IP address. Case-insensitive regular expression (PCRE).	10.10 ^192.168.1.2$
mac	string	No	Asset MAC address. Case-insensitive regular expression (PCRE).	^00:0a:95:9d:68:16$

Response

HTTP code: 200

Format: JSON

```
type Response []Asset

type Asset struct {
ID string `json:"id"`
TenantID string `json:"tenantID"`
TenantName string `json:"tenantName"`
Name string `json:"name"`
FQDN string `json:"fqdn"`
IPAddresses []string `json:"ipAddresses"`
MACAddresses []string `json:"macAddresses"`
Owner string `json:"owner"`
OS *OS `json:"os"`
Software []Software `json:"software"`
Vulnerabilities []Vulnerability `json:"vulnerabilities"`
KICSRisks []*assets.KICSRisk `json:"kicsVulns"`
KSC *KSCFields `json:"ksc"`
Created string `json:"created"`
Updated string `json:"updated"`
}

type KSCFields struct {
NAgentID string `json:"nAgentID"`
KSCInstanceID string `json:"kscInstanceID"`
KSCMasterHostname string `json:"kscMasterHostname"`
LastVisible string `json:"lastVisible"`
}
```

```
type OS struct {
Name string `json:"name"`
Version uint64 `json:"version"`
}

type Software struct {
Name string `json:"name"`
Version string `json:"version"`
Vendor string `json:"vendor"`
}

type Vulnerability struct {
KasperskyID string `json:"kasperskyID"`
ProductName string `json:"productName"`
DescriptionUrl string `json:"descriptionUrl"`
RecommendedMajorPatch string `json:"recommendedMajorPatch"`
RecommendedMinorPatch string `json:"recommendedMinorPatch"`
SeverityStr string `json:"severityStr"`
Severity uint64 `json:"severity"`
CVE []string `json:"cve"`
ExploitExists bool `json:"exploitExists"`
MalwareExists bool `json:"malwareExists"`
}

type assets.KICSRisk struct {
ID int64 `json:"id"`
Name string `json:"name"`
Category string `json:"category"`
Description string `json:"description"`
DescriptionUrl string `json:"descriptionUrl"`
Severity int `json:"severity"`
Cvss float64 `json:"cvss"`
}

type CustomFields struct {
ID string `json:"id"`
Name string `json:"name"`
Value string `json:"value"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Invalid value of the "page" parameter	invalid query parameter value	page
500	Any other internal errors	variable	variable

Importing assets

Details on identifying, creating, and updating assets

Assets are imported according to the asset data merging rules.

POST /api/v1/assets/import

Bulk creation or update of assets.

If the FQDN of an asset is specified, it acts as the unique ID of the asset within the tenant. If more than one FQDN is specified, the first FQDN from the specified array of FQDNs is used. If no FQDN is specified, the first IP address in the specified array of addresses is used to identify the asset. If the asset name is not specified, either FQDN or the first IP address is used as the name. Assets imported from KSC cannot be updated, therefore, FQDN conflicts may occur during the import process if a KSC asset with a the same FQDN already exists in the tenant. Such conflicts prevent the processing of the conflicting asset, but do not prevent the processing of other assets specified in the request body. Allows you to populate custom fields by uuid from the assetsCustomFields settings.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Request body

Format: JSON

```
type Request struct {
TenantID string `json:"tenantID"`
    Assets []Asset `json:"assets"`
}

type Asset struct {
Name string `json:"name"`
FQDN string `json:"fqdn"`
IPAddresses []string `json:"ipAddresses"`
MACAddresses []string `json:"macAddresses"`
Owner string `json:"owner"`
OS *OS `json:"os"`
Software []Software `json:"software"`
Vulnerabilities []Vulnerability `json:"vulnerabilities"`
CustomFields []Software `json:"customFields"`
}

type OS struct {
Name string `json:"name"`
Version uint64 `json:"version"`
}

type Software struct {
Name string `json:"name"`
Version string `json:"version"`
Vendor string `json:"vendor"`
}

type Vulnerability struct {
KasperskyID string `json:"kasperskyID"`
ProductName string `json:"productName"`
DescriptionUrl string `json:"descriptionUrl"`
RecommendedMajorPatch string `json:"recommendedMajorPatch"`
RecommendedMinorPatch string `json:"recommendedMinorPatch"`
SeverityStr string `json:"severityStr"`
Severity uint64 `json:"severity"`
CVE []string `json:"cve"`
ExploitExists bool `json:"exploitExists"`
MalwareExists bool `json:"malwareExists"`
}

type CustomFields struct {
ID string `json:"id"`
Name string `json:"name"`
Value string `json:"value"`
}
```

Request mandatory fields

Name	Data type	Mandatory	Description	Value example
TenantID	string	Yes	Tenant ID	00000000-0000-0000-0000-000000000000
assets	[]Asset	Yes	Array of imported assets	

Asset mandatory fields

Name	Data type	Mandatory	Description	Value example
fqdn	string	If the ipAddresses array is not specified	Asset FQDN. You can specify multiple values separated by commas. It is recommended that you specify the FQDN and not just the host name. Priority indicator for asset identification.	my-asset-1.example.com my-asset-1
ipAddresses	[]string	If FQDN is not specified	Array of IP addresses for the asset. IPv4 or IPv6. The first element of the array is used as a secondary indicator for asset identification.	["192.168.1.1", "192.168.2.2"] ["2001:0db8:85a3:0000:0000:8a2e:0370:7334"]

Response

HTTP code: 200

Format: JSON

```
type Response struct {
InsertedIDs map[int64]interface{} `json:"insertedIDs"`
UpdatedCount uint64 `json:"updatedCount"`
Errors []ImportError `json:"errors"`
}

type ImportError struct {
Index uint64 `json:"index"`
Message string `json:"message"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Tenant ID is not specified	tenantID required	
400	Attempt to import assets into the shared tenant	import into shared tenant not allowed	
400	Not a single asset was specified in the request body	at least one asset required	
400	None of the mandatory fields is specified	one of fields required	asset[<index>]: fqdn, ipAddresses
400	Invalid FQDN	invalid value	asset[<index>].fqdn
400	Invalid IP address	invalid value	asset[<index>].ipAddresses[<index>]
400	IP address is repeated	duplicated value	asset[<index>].ipAddresses
400	Invalid MAC address	invalid value	asset[<index>].macAddresses[<index>]
400	MAC address is repeated	duplicated value	asset[<index>].macAddresses
403	The user does not have the required role in the specified tenant	access denied	
404	The specified tenant was not found	tenant not found	
406	The specified tenant was disabled	tenant disabled	
500	Any other internal errors	variable	variable

Deleting assets

POST /api/v1/assets/delete

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Request body

Format: JSON

Name	Data type	Mandatory	Description	Value example
TenantID	string	Yes	Tenant ID	00000000-0000-0000-0000-000000000000
ids	[]string	If neither the ipAddresses array nor the FQDNs are specified	List of asset IDs	["00000000-0000-0000-0000-000000000000"]
fqdns	[]string	If neither the ipAddresses array nor the IDs are specified	Array of asset FQDNs	["my-asset-1.example.com", "my-asset-1"]
ipAddresses	[]string	If neither the IDs nor FQDNs are specified	Array of main IP addresses of the asset.	["192.168.1.1", "2001:0db8:85a3:0000:0000:8a2e:0370:7334"]

Response

HTTP code: 200

Format: JSON

```
type Response struct {
DeletedCount uint64 `json:"deletedCount"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Tenant ID is not specified	tenantID required	
400	Attempt to delete an asset from the shared tenant	delete from shared tenant not allowed	
400	None of the mandatory fields is specified	one of fields required	ids, fqdns, ipAddresses
400	Invalid FQDN specified	invalid value	fqdns[<index>]
400	Invalid IP address specified	invalid value	ipAddresses[<index>]
403	The user does not have the required role in the specified tenant	access denied	
404	The specified tenant was not found	tenant not found	
406	The specified tenant was disabled	tenant disabled	

500	Any other internal errors	variable	variable

Searching events

POST /api/v1/events

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, Access to NCIRCC, Access to CII.

Request body

Format: JSON

Request

Name	Data type	Mandatory	Description	Value example
period	Period	Yes	Search period	

sql	string	Yes	SQL query	SELECT * FROM events WHERE Type = 3 ORDER BY Timestamp DESC LIMIT 1000
				SELECT sum(BytesOut) as TotalBytesSent, SourceAddress FROM events WHERE DeviceVendor = 'netflow' GROUP BY SourceAddress LIMIT 1000
				SELECT count(Timestamp) as TotalEvents FROM events LIMIT 1

ClusterID	string	No, if the cluster is the only one	Storage cluster ID. You can find it by requesting a list of services with kind = storage. The cluster ID will be in the resourceID field.	00000000-0000-0000-0000-000000000000
rawTimestamps	bool	No	Display timestamps in their current format—Milliseconds since EPOCH. False by default.	true or false
emptyFields	bool	No	Display empty fields for normalized events. False by default.	true or false

Period

Name	Data type	Mandatory	Description	Value example
from	string	Yes	Lower bound of the period in RFC3339 format. Timestamp >= <from>	2021-09-06T00:00:00Z (UTC) 2021-09-06T00:00:00.000Z (UTC, including milliseconds)

				2021-09-06T00:00:00Z+00:00 (MSK)
to	string	Yes	Upper bound of the period in RFC3339 format. Timestamp <= <to>	2021-09-06T00:00:00Z (UTC) 2021-09-06T00:00:00.000Z (UTC, including milliseconds) 2021-09-06T00:00:00Z+00:00 (MSK)

Response

HTTP code: 200

Format: JSON

Result of executing the SQL query

Possible errors

HTTP code	Description	message field value	details field value
400	The lower bounds of the range is not specified	period.from required	
400	The lower bounds of the range is in an unsupported format	cannot parse period.from	variable
400	The lower bounds of the range is equal to zero	period.from cannot be 0	
400	The upper bounds of the range is not specified	period.to required	
400	The upper bounds of the range is in an unsupported format	cannot parse period.to	variable
400	The upper bounds of the range is equal to zero	period.to cannot be 0	
400	The lower bounds of the range is greater than the upper bounds	period.from cannot be greater than period.to	
400	Invalid SQL query	invalid sql	variable
400	An invalid table appears in the SQL query	the only valid table is `events`	
400	The SQL query lacks a LIMIT	sql: LIMIT required	
400	The LIMIT in the SQL query exceeds the maximum (1000)	sql: maximum LIMIT is 1000	
404	Storage cluster not found	cluster not found	
406	The clusterID parameter was not specified, and many clusters were registered in KUMA	multiple clusters found, please provide clusterID	
500	No available cluster nodes	no nodes available	
50x	Any other internal errors	event search failed	variable

Viewing information about the cluster

GET /api/v1/events/clusters

Access: The main tenant clusters are accessible to all users.

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
page	number	No	Page number. Starts with 1. The page size is 250 entries. If the parameter is not specified, the default value is 1.	1
id	string	No	Cluster ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied	00000000-0000-0000-0000-000000000000
TenantID	string	No	Tenant ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied. If the user does not have the required role in the specified tenant, then this tenant is ignored.	00000000-0000-0000-0000-000000000000
name	string	No	Cluster name. Case-insensitive regular expression (PCRE).	cluster ^My cluster$

Response

HTTP code: 200

Format: JSON

```
type Response []Cluster

type Cluster struct {
ID string `json:"id"`
Name string `json:"name"`
TenantID string `json:"tenantID"`
TenantName string `json:"tenantName"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Invalid value of the "page" parameter	invalid query parameter value	page
500	Any other internal errors	variable	variable

Resource search

GET /api/v1/resources

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Access to shared resources.

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
page	number	No	Page number. Starts with 1. The page size is 250 entries. If the parameter is not specified, the default value is 1.	1
id	string	No	Resource ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied.	00000000-0000-0000-0000-000000000000
TenantID	string	No	Resource tenant ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied. If the user does not have the required role in the	00000000-0000-0000-0000-000000000000

			specified tenant, then this tenant is ignored.	
name	string	No	Resource name. Case-insensitive regular expression (PCRE).	resource ^My resource$
kind	string	No	Resource type. If the parameter is specified several times, then a list is generated and the logical OR operator is applied	collector, correlator, storage, activeList, aggregationRule, con enrichmentRule, destination, filter, normalizer, responseRule, s

Response

HTTP code: 200

Format: JSON

```
type Response []Resource

type Resource struct {
ID string `json:"id"`
Kind string `json:"kind"`
Name string `json:"name"`
Description string `json:"description"`
TenantID string `json:"tenantID"`
TenantName string `json:"tenantName"`
UserID string `json:"userID"`
UserName string `json:"userName"`
Created string `json:"created"`
Updated string `json:"updated"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Invalid value of the "page" parameter	invalid query parameter value	page
400	Invalid value of the "kind" parameter	invalid kind	<kind>

500	Any other internal errors	variable	variable

Loading resource file

POST /api/v1/resources/upload

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Request body

Encrypted contents of the resource file in binary format.

Response

HTTP code: 200

Format: JSON

File ID. It should be specified in the body of requests for viewing the contents of the file and for importing resources.

```
type Response struct {
ID string `json:"id"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	The file size exceeds the maximum allowable (64 MB)	maximum file size is 64 MB	
403	The user does not have the required roles in any of the tenants	access denied	
500	Any other internal errors	variable	variable

Viewing the contents of a resource file

POST /api/v1/resources/toc

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Request body

Format: JSON

Name	Data type	Mandatory	Description	Value example
fileID	string	Yes	The file ID obtained as a result of loading the resource file.	00000000-0000-0000-0000-000000000000
password	string	Yes	Resource file password.	SomePassword!88

Response

HTTP code: 200

Format: JSON

File version, list of resources, categories, and folders.

The ID of the retrieved resources must be used when importing.

```
type Package struct {
Version string `json:"version"`
AssetCategories []*categories.Category `json:"assetCategories"`
Folders []*folders.Folder `json:"folders"`
Resources []*resources.ExportedResource `json:"resources"`
}
```

Importing resources

POST /api/v1/resources/import

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Request body

Name	Data type	Mandatory	Description	Value example
fileID	string	Yes	The file ID obtained as a result of loading the resource file.	00000000-0000-0000-0000-000000000000
password	string	Yes	Resource file password.	SomePassword!88
TenantID	string	Yes	ID of the target tenant	00000000-0000-0000-0000-000000000000
actions	map[string]uint8	Yes	Mapping of the resource ID to the action that must be taken in relation to it.	0—do not import (used when resolving conflicts)

				1—import (should initially be assigned to each resource)
				2—replace (used when resolving conflicts)
				`{`
				`"00000000-0000-0000-0000-000000000000": 0,`
				`"00000000-0000-0000-0000-000000000001": 1,`
				`"00000000-0000-0000-0000-000000000002": 2,`
				`}`

Response

HTTP code	Body
204	

409	The imported resources conflict with the existing ones by ID. In this case, you need to repeat the import operation while specifying the following actions for these resources:
	0—do not import
	2—replace
	`type ImportConflictsError struct {`
	``HardConflicts []string `json:"conflicts"` ``
	`}`

Exporting resources

POST /api/v1/resources/export

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Access to shared resources.

Request body

Format: JSON

Name	Data type	Mandatory	Description	Value example
ids	[]string	Yes	Resource IDs to be exported	["00000000-0000-0000-0000-000000000000"]
password	string	Yes	Exported resource file password	SomePassword!88
TenantID	string	Yes	ID of the tenant that owns the exported resources	00000000-0000-0000-0000-000000000000

Response

HTTP code: 200

Format: JSON

ID of the file with the exported resources. It should be used in a request to download the resource file.

```
type ExportResponse struct {
FileID string `json:"fileID"`
}
```

Downloading the resource file

GET /api/v1/resources/download/<id>

Here "id" is the file ID obtained as a result of executing a resource export request.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Response

HTTP code: 200

Encrypted contents of the resource file in binary format.

Possible errors

HTTP code	Description	message field value	details field value
400	File ID not specified	route parameter required	id
400	The file ID is not a valid UUID	id is not a valid UUID	
403	The user does not have the required roles in any of the tenants	access denied	
404	File not found	file not found	

406	The file is a directory	not regular file	
500	Any other internal errors	variable	variable

Search for services

GET /api/v1/services

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
page	number	No	Page number. Starts with 1. The page size is 250 entries. If the parameter is not specified, the default value is 1.	1
id	string	No	Service ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied.	00000000-0000-0000-0000-000000000000
TenantID	string	No	Service tenant ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied. If the user does not have the required role in the specified tenant, then this tenant is ignored.	00000000-0000-0000-0000-000000000000
name	string	No	Service name. Case-insensitive regular expression (PCRE).	service ^My service$
kind	string	No	Service type. If the parameter is specified several times, then a list is generated and the logical OR operator is applied.	collector, correlator, storage, agent
fqdn	string	No	Service FQDN. Case-insensitive regular expression (PCRE).	hostname ^hostname.example.com$
paired	bool	No	Display only those services that executed the first start. If the parameter is present in the URL query, then its value is assumed to be true. The values specified by the user are ignored. Example: /api/v1/services?paired	

Response

HTTP code: 200

Format: JSON

```
type Response []Service

type Service struct {
ID string `json:"id"`
TenantID string `json:"tenantID"`
TenantName string `json:"tenantName"`
ResourceID string `json:"resourceID"`
Kind string `json:"kind"`
Name string `json:"name"`
Address string `json:"address"`
FQDN string `json:"fqdn"`
Status string `json:"status"`
Warning string `json:"warning"`
APIPort string `json:"apiPort"`
Uptime string `json:"uptime"`
Version string `json:"version"`
Created string `json:"created"`
Updated string `json:"updated"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Invalid value of the "page" parameter	invalid query parameter value	page
400	Invalid value of the "kind" parameter	invalid kind	<kind>
500	Any other internal errors	variable	variable

Tenant search

GET /api/v1/tenants

Only tenants available to the user are displayed.

Access: General administrator, Administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, Interaction with NCIRCC, Access to CII, Access to shared resources.

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
page	number	No	Page number. Starts with 1. The page size is 250 entries. If the parameter is not specified, the default value is 1.	1

id	string	No	Tenant ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied.	00000000-0000-0000-0000-000000000000
name	string	No	Tenant name. Case-insensitive regular expression (PCRE).	tenant ^My tenant$
main	bool	No	Only display the main tenant. If the parameter is present in the URL query, then its value is assumed to be true. The values specified by the user are ignored. Example: /api/v1/tenants?main	

Response

HTTP code: 200

Format: JSON

```
type Response []Tenant

type Tenant struct {
    ID          string `json:"id"`
    Name        string `json:"name"`
    Main        bool   `json:"main"`
    Description string `json:"description"`
    EPS         uint64 `json:"eps"`
    EPSLimit    uint64 `json:"epsLimit"`
    Created     string `json:"created"`
    Updated     string `json:"updated"`
Shared   bool   `json:"shared"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Invalid value of the "page" parameter	invalid query parameter value	page
500	Any other internal errors	variable	variable

View token bearer information

GET /api/v1/users/whoami

Response

HTTP code: 200

Format: JSON

```
type Response struct {
ID string `json:"id"`
Name string `json:"name"`
Login string `json:"login"`
Email string `json:"email"`
Tenants []TenantAccess `json:"tenants"`
}

type TenantAccess struct {
ID string `json:"id"`
Name string `json:"name"`
Role string `json:"role"`
}
```

Dictionary updating in services

POST /api/v1/dictionaries/update

You can update only dictionaries in dictionary resources of the table type.

Access: General administrator, Tenant administrator, Tier 2 analyst for all tenants except the Shared tenant; General administrator for the Shared tenant; Tier 1 analyst — only their own.

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
dictionaryID	string	Yes	ID of the dictionary that will be updated.	00000000-0000-0000-0000-000000000000

The update affects all services where the specified dictionary is used. If an update in one of the services ends with an error, this does not interrupt updates in the other services.

Request body

Multipart field name	Data type	Mandatory	Description	Value example
file	CSV file	Yes	The request contains a CSV file. Data of the existing dictionary is being replaced with data from this file. The first line of the CSV file containing the column names must not be changed.	key columns,column1,column2 key1,k1col1,k1col2 key2,k2col1,k2col2

Response

HTTP code: 200

Format: JSON

```
type Response struct {
ServicesFailedToUpdate []UpdateError `json:"servicesFailedToUpdate"`
}
type UpdateError struct {
ID string `json:"id"`
Err error `json:"err"`
}
```

Returns only errors for services in which the dictionaries have not been updated.

Possible errors

HTTP code	Description	message field value	details field value
400	Invalid request body	request body decode failed	Error
400	Null count of dictionary lines	request body required	
400	Dictionary ID not specified	invalid value	dictionaryID
400	Incorrect value of dictionary line	invalid value	rows or rows[i]
400	Dictionary with the specified ID has an invalid type (not table)	can only update table dictionary	
400	Attempt to change dictionary columns	columns must not change with update	
403	No access to requested resource	access denied	
404	Service not found	service not found	
404	Dictionary not found	dictionary not found	Service ID
500	Any other internal errors	variable	variable

Dictionary retrieval

GET /api/v1/dictionaries

You can get only dictionaries in dictionary resources of the table type.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
dictionaryID	string	Yes	ID of the dictionary that will be received	00000000-0000-0000-0000-000000000000

Response

HTTP code: 200

Format: text/plain; charset=utf-8

A CSV file is returned with the dictionary data in the response body.

Viewing custom fields of the assets

GET /api/v1/settings/id/:id

The user can view a list of custom fields made by the KUMA user in the application web interface.

A custom field is a bucket for entering text. If necessary, the default value and the mask can be used to validate the entered text in the following format: https://pkg.go.dev/regexp/syntax. All forward slash characters in the mask must be shielded.

Access: General administrator, Main tenant administrator; Tier 1 or Tier 2 analyst of the Main tenant (must have rights to the requested setting).

Query parameters

Name	Data type	Mandatory	Description	Value example
id	string	Yes	Configuration ID of the custom fields	00000000-0000-0000-0000-000000000000

Response

HTTP code: 200

Format: JSON

```
type Settings struct {
ID string `json:"id"`
TenantID string `json:"tenantID"`
TenantName string `json:"tenantName"`
Kind string `json:"kind"`
UpdatedAt int64 `json:"updatedAt"`
CreatedAt int64 `json:"createdAt"`
Disabled bool `json:"disabled"`
CustomFields []*CustomField `json:"customFields"`
}

type CustomField struct {
ID string `json:"id"`
Name string `json:"name"`
Default string `json:"default"`
Mask string `json:"mask"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
404	Parameters not found: invalid ID or parameters are missing	Not found in database	null
500	Any other internal errors	variable	variable

Creating a backup of the KUMA Core

GET /api/v1/system/backup

Access: General administrator.

The request has no parameters.

The tar.gz archive containing the backup copy of the KUMA Core is returned in response to the request. The backup copy is not saved on the host where the Core is installed. The certificates are included in the backup copy.

If the operation is successful, an audit event is generated with the following parameters:

- `DeviceAction = "Core backup created"`
- `SourceUserID = "<user-login>"`

You can restore the KUMA Cores from a backup using the following API request: `POST /api/v1/system/restore`.

Restoring the KUMA Core from the backup

POST /api/v1/system/restore

Access: General administrator.

The request has no parameters.

The request body must contain an archive with the backup copy of the KUMA Core, obtained as a result of the following API request execution: `GET /api/v1/system/backup`.

After receiving the archive with the backup copy, KUMA performs the following actions:

1. Unpacks the archive with the backup copy of the KUMA Core to a temporary directory.
2. Compares the current KUMA version with the backup KUMA version. Data may only be restored from a backup if it is restored to the KUMA of the same version as the backup one.

If the versions match, an audit event is generated with the following parameters:

- `DeviceAction = "Core restore scheduled"`
- `SourceUserID = "<name of the user who initiated KUMA restore from a backup copy>"`

3. If the versions match, data is restored from the backup copy of the KUMA Core.
4. The temporary directory is deleted, and KUMA starts normally.

   The "WARN: restored from backup" entry is added to the KUMA Core log.

Viewing the list of context tables in the correlator

GET /api/v1/contextTables
The target correlator must be running.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
correlatorID	string	Yes	Correlator service ID	00000000-0000-0000-0000-000000000000

Response

HTTP code: 200

Format: JSON

```
type Response []ContextTableInfo

type ContextTableInfo struct {
ID string `json:"id"`
Name string `json:"name"`
Dir string `json:"dir"`
Records uint64 `json:"records"`
WALSize uint64 `json:"walSize"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Correlator service ID is not specified.	query parameter required	correlatorID
403	The user does not have the required role in the	access denied	-

	correlator tenant.		
404	The service with the specified ID (correlatorID) was not found.	service not found	-
406	The service with the specified ID (correlatorID) is not a correlator.	service is not correlator	-
406	The correlator did not execute the first start.	service not paired	-
406	The tenant of the correlator is disabled.	tenant disabled	-
50x	Failed to gain access to the correlator API.	correlator API request failed	variable
500	Failed to decode the body of the response received from the correlator.	correlator response decode failed	variable
500	Any other internal error.	variable	variable

Importing records into a context table

POST /api/v1/contextTables/import

The target correlator must be running.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst (can import data into any correlator table of an accessible tenant, even if the context table was created in the Shared tenant).

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
correlatorID	string	Yes	Correlator service ID	00000000-0000-0000-0000-000000000000
contextTableID	string	If contextTableName is not specified	Context table ID	00000000-0000-0000-0000-000000000000
contextTableName	string	If contextTableID is not specified	Name of the context table	Attackers
format	string	Yes	Format of imported entries	CSV, TSV, internal
clear	bool	No	Clear the context table before importing. If the parameter is present in the URL query, its value is assumed to be true. The values specified by the user are ignored.	/api/v1/contextTables/import?clear

Request body

Format	Contents
CSV	The first row is the header, which lists the comma-separated fields. The rest of the rows are the comma-separated values corresponding to the fields in the header. The number of fields in each row must be the same, and it must match the number of fields in the schema of the context table. List field values are separated by the "|" character. For example, the value of a list of integers might be 1|2|3.
TSV	The first row is the header, which lists the TAB-separated fields. The rest of the rows are the TAB-separated values corresponding to the fields in the header. The number of fields in each row must be the same, and it must match the number of fields in the schema of the context table. List field values are separated by the "|" character.
internal	Each line contains one individual JSON object. Data in the 'internal' format can be obtained by exporting the contents of the context table from the correlator in the KUMA web console.

Response

HTTP code: 204

Possible errors

HTTP code	Description	message field value	details field value
400	Correlator service ID is not specified.	query parameter required	correlatorID
400	Neither the contextTableID parameter nor the contextTableName parameter is specified	one of query parameters required	contextTableID, contextTableName
400	The 'format' parameter is not specified	query parameter required	format
400	The 'format' parameter is invalid	invalid query parameter value	format
400	The request body has zero length	request body required	-
400	Error parsing the request body, including the non-conformance of the field names and types of the record being imported with the schema of the context table.	correlator API request failed	variable
403	The user does not have the required role in the correlator tenant.	access denied	-
404	The service with the specified ID (correlatorID) was not found.	service not found	-
404	The context table was not found.	context table not found	-
406	The service with the specified ID (correlatorID) is not a correlator.	service is not correlator	-
406	The correlator did not execute the first start.	service not paired	-
406	The tenant of the correlator is disabled.	tenant disabled	-

406	More than one context table found by a search for contextTableName.	more than one matching context tables found	-
50x	Failed to gain access to the correlator API.	correlator API request failed	variable
500	Error preparing data for importing into the correlator service.	context table process import request failed	variable
500	Any other internal error.	variable	variable

Exporting records from a context table

GET /api/v1/contextTables/export

The target correlator must be running.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
correlatorID	string	Yes	Correlator service ID	00000000-0000-0000-0000-000000000000
contextTableID	string	If contextTableName is not specified	Context table ID	00000000-0000-0000-0000-000000000000
contextTableName	string	If contextTableID is not specified	Name of the context table	Attackers

Response

HTTP code: 200

Format: application/octet-stream

Body: exported context table data, in the 'internal' format: each row contains one individual JSON object.

Possible errors

HTTP code	Description	message field value	details field value
400	Correlator service ID is not specified.	query parameter required	correlatorID
400	Neither the contextTableID parameter nor the contextTableName parameter is specified	one of query parameters required	contextTableID, contextTableName

403	The user does not have the required role in the correlator tenant.	access denied	-
404	The service with the specified ID (correlatorID) was not found.	service not found	-
404	The context table was not found.	context table not found	-
406	The service with the specified ID (correlatorID) is not a correlator.	service is not correlator	-
406	The correlator did not execute the first start.	service not paired	-
406	The tenant of the correlator is disabled.	tenant disabled	-
406	More than one context table found by a search for contextTableName.	more than one matching context tables found	-
50x	Failed to gain access to the correlator API.	correlator API request failed	variable
500	Any other internal error.	variable	variable

REST API v2 operations

Description of available requests and responses.

Viewing a list of active lists on the correlator

GET /api/v2/activeLists

The target correlator must be running.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Query parameters

Name	Data type	Mandatory	Description	Value example
correlatorID	string	Yes	Correlator service ID	00000000-0000-0000-0000-000000000000

Response

HTTP code: 200

Format: JSON

```
type Response []ActiveListInfo

type ActiveListInfo struct {
    ID      string `json:"id"`
    Name    string `json:"name"`
    Dir     string `json:"dir"`
    Records uint64 `json:"records"`
    WALSize uint64 `json:"walSize"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Correlator service ID is not specified	query parameter required	correlatorID
403	The user does not have the required role in the correlator tenant	access denied	-
404	The service with the specified identifier (correlatorID) was not found	service not found	-
406	The service with the specified ID (correlatorID) is not a correlator	service is not correlator	-
406	The correlator did not execute the first start	service not paired	-
406	The correlator tenant is disabled	tenant disabled	-
50x	Failed to access the correlator API	correlator API request failed	variable
500	Failed to decode the response body received from the correlator	correlator response decode failed	variable
500	Any other internal errors	variable	variable

Import entries to an active list

POST /api/v2/activeLists/import

The target correlator must be running.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst (can import data into any correlator list of an accessible tenant, even if the active list was created in the Shared tenant).

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
correlatorID	string	Yes	Correlator service ID	00000000-0000-0000-0000-000000000000
activeListID	string	If activeListName is not specified	Active list ID	00000000-0000-0000-0000-000000000000

activeListName	string	If activeListID is not specified	Active list name	Attackers
format	string	Yes	Format of imported entries	CSV, TSV, internal
keyField	string	For the CSV and TSV formats only	The name of the field in the header of the CSV or TSV file that will be used as the key field of the active list record. The values of this field must be unique	ip
clear	bool	No	Clear the active list before importing. If the parameter is present in the URL query, then its value is assumed to be true. The values specified by the user are ignored.	/api/v2/activeLists/import?clear

Request body

Format	Contents
CSV	The first line is the header, which lists the comma-separated fields. The rest of the lines are the values corresponding to the comma-separated fields in the header. The number of fields in each line must be the same.
TSV	The first line is the header, which lists the TAB-separated fields. The remaining lines are the values corresponding to the TAB-separated fields in the header. The number of fields in each line must be the same.
internal	Each line contains one individual JSON object. Data in the internal format can be received by exporting the contents of the active list from the correlator in the KUMA web console.

Response

HTTP code: 204

Possible errors

HTTP code	Description	message field value	details field value
400	Correlator service ID is not specified	query parameter required	correlatorID
400	Neither the activeListID parameter nor the activeListName parameter is specified	one of query parameters required	activeListID, activeListName
400	The format parameter is not specified	query parameter required	format
400	The format parameter is invalid	invalid query parameter value	format
400	The keyField parameter is not specified	query parameter required	keyField

400	The request body has a zero-length	request body required	-
400	The CSV or TSV file does not contain the field specified in the keyField parameter	correlator API request failed	variable
400	Request body parsing error	correlator API request failed	variable
403	The user does not have the required role in the correlator tenant	access denied	-
404	The service with the specified identifier (correlatorID) was not found	service not found	-
404	No active list was found	active list not found	-
406	The service with the specified ID (correlatorID) is not a correlator	service is not correlator	-
406	The correlator did not execute the first start	service not paired	-
406	The correlator tenant is disabled	tenant disabled	-
406	A search was performed using the name of the active list (activeListName), and more than one active list was found	more than one matching active lists found	-
50x	Failed to access the correlator API	correlator API request failed	variable
500	Failed to decode the response body received from the correlator	correlator response decode failed	variable
500	Any other internal errors	variable	variable

Searching alerts

GET /api/v2/alerts

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, Interaction with NCIRCC, Access to CII.

Query parameters

Name	Data type	Mandatory	Description	Value example
page	number	No	Page number. Starts with 1. The page size is 250 entries. If the parameter is not specified, the default value is 1.	1
id	string	No	Alert ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied.	00000000-0000-0000-0000-000000000000
TenantID	string	No	Alert tenant ID. If the parameter is specified several times, then a list is generated and the logical OR operator is	00000000-0000-0000-0000-000000000000

			applied. If the user does not have the required role in the specified tenant, then this tenant is ignored.	
name	string	No	Alert name. Case-insensitive regular expression (PCRE).	alert ^My alert$
timestampField	string	No	The name of the alert field that is used to perform sorting (DESC) and search by period (from – to). lastSeen by default.	lastSeen, firstSeen
from	string	No	Lower bound of the period in RFC3339 format. <timestampField> >= <from>	2021-09-06T00:00:00Z (UTC) 2021-09-06T00:00:00.000Z (UTC, including milliseconds) 2021-09-06T00:00:00Z+00:00 (MSK)
to	string	No	Upper bound of the period in RFC3339 format. <timestampField> <= <to>	2021-09-06T00:00:00Z (UTC) 2021-09-06T00:00:00.000Z (UTC, including milliseconds) 2021-09-06T00:00:00Z+00:00 (MSK)
status	string	No	Alert status. If the parameter is specified several times, then a list is generated and the logical OR operator is applied.	new, assigned, escalated, closed
withEvents	bool	No	Include normalized KUMA events associated with found alerts in the response. If the parameter is present in the URL query, then its value is assumed to be true. The values specified by the user are ignored. Example: /api/v1/alerts?withEvents	-
withAffected	bool	No	Include information about the assets and accounts associated with the found alerts in the report. If the parameter is present in the URL query, then its value is assumed to be true. The values specified by the user are ignored. Example: /api/v1/alerts?withAffected	-

Response

HTTP code: 200

Format: JSON

```
type Response []Alert
type Alert struct {
    ID                string            `json:"id"`
    TenantID          string            `json:"tenantID"`
    TenantName        string            `json:"tenantName"`
    Name              string            `json:"name"`
    CorrelationRuleID string            `json:"correlationRuleID"`
    Priority          string            `json:"priority"`
    Status            string            `json:"status"`
    FirstSeen         string            `json:"firstSeen"`
    LastSeen          string            `json:"lastSeen"`
    Assignee          string            `json:"assignee"`
    ClosingReason     string            `json:"closingReason"`
    Overflow          bool              `json:"overflow"`
    Events            []NormalizedEvent `json:"events"`
    AffectedAssets    []AffectedAsset   `json:"affectedAssets"`
    AffectedAccounts  []AffectedAccount `json:"affectedAccounts"`
}

type NormalizedEvent map[string]interface{}

type AffectedAsset struct {
    ID               string          `json:"id"`
    TenantID         string          `json:"tenantID"`
    TenantName       string          `json:"tenantName"`
    Name             string          `json:"name"`
    FQDN             string          `json:"fqdn"`
    IPAddresses      []string        `json:"ipAddresses"`
    MACAddresses     []string        `json:"macAddresses"`
    Owner            string          `json:"owner"`
    OS               *OS             `json:"os"`
    Software         []Software      `json:"software"`
    Vulnerabilities  []Vulnerability `json:"vulnerabilities"`
    KSC              *KSCFields      `json:"ksc"`
    Created          string          `json:"created"`
    Updated          string          `json:"updated"`
}

type OS struct {
    Name    string `json:"name"`
    Version uint64 `json:"version"`
}

type Software struct {
    Name    string `json:"name"`
    Version string `json:"version"`
    Vendor  string `json:"vendor"`
}

type Vulnerability struct {
    KasperskyID           string   `json:"kasperskyID"`
    ProductName           string   `json:"productName"`
    DescriptionURL        string   `json:"descriptionURL"`
    RecommendedMajorPatch string   `json:"recommendedMajorPatch"`
    RecommendedMinorPatch string   `json:"recommendedMinorPatch"`
    SeverityStr           string   `json:"severityStr"`
    Severity              uint64   `json:"severity"`
    CVE                   []string `json:"cve"`
    ExploitExists         bool     `json:"exploitExists"`
    MalwareExists         bool     `json:"malwareExists"`
}

type AffectedAccount struct {
    Name             string `json:"displayName"`
    CN               string `json:"cn"`
    DN               string `json:"dn"`
    UPN              string `json:"upn"`
    SAMAccountName   string `json:"sAMAccountName"`
    Company          string `json:"company"`
    Department       string `json:"department"`
    Created          string `json:"created"`
    Updated          string `json:"updated"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Invalid value of the "page" parameter	invalid query parameter value	page
400	Invalid value of the "status" parameter	invalid status	<status>
400	Invalid value of the "timestampField" parameter	invalid timestamp field	-
400	Invalid value of the "from" parameter	cannot parse from	variable
400	Invalid value of the "to" parameter	cannot parse to	variable
400	The value of the "from" parameter is greater than the value of the "to" parameter	from cannot be greater than to	-
500	Any other internal errors	variable	variable

Closing alerts

POST /api/v2/alerts/close

The target correlator must be running.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, Interaction with NCIRCC, Access to CII.

Request body

Format: JSON

Name	Data type	Mandatory	Description	Value example

id	string	Yes	Alert ID	00000000-0000-0000-0000-000000000000
reason	string	Yes	Reason for closing the alert	responded, incorrect data, incorrect correlation rule

Response

HTTP code: 204

Possible errors

HTTP code	Description	message field value	details field value
400	Alert ID is not specified	id required	-
400	The reason for closing the alert is not specified	reason required	-
400	Invalid value of the "reason" parameter	invalid reason	-
403	The user does not have the required role in the alert tenant	access denied	-
404	Alert not found	alert not found	-
406	Alert tenant disabled	tenant disabled	-
406	Alert already closed	alert already closed	-
500	Any other internal errors	variable	variable

Searching assets

GET /api/v2/assets

Information about the software of assets from KSC is not stored in KUMA and is not shown in the response.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, Access to NCIRCC, Access to CII.

The "Access to shared resources" role is issued only for the Shared tenant; this tenant cannot have any assets, but it has categories. For this role, nothing is returned in the response.

Query parameters

Name	Data type	Mandatory	Description	Value example
page	number	No	Page number. Starts with 1. The page size is 250 entries. If the parameter is not specified, the default value is 1.	1

id	string	No	Asset ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied.	00000000-0000-0000-0000-000000000000
TenantID	string	No	Asset tenant ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied. If the user does not have the required role in the specified tenant, then this tenant is ignored.	00000000-0000-0000-0000-000000000000
name	string	No	Asset name. Case-insensitive regular expression (PCRE).	asset ^My asset$
fqdn	string	No	Asset FQDN. Case-insensitive regular expression (PCRE).	example.com
ip	string	No	Asset IP address. Case-insensitive regular expression (PCRE).	10.10 ^192.168.1.2$
mac	string	No	Asset MAC address. Case-insensitive regular expression (PCRE).	^00:0a:95:9d:68:16$

Response

HTTP code: 200

Format: JSON

```
type Response []Asset

type Asset struct {
ID string `json:"id"`
TenantID string `json:"tenantID"`
TenantName string `json:"tenantName"`
Name string `json:"name"`
FQDN string `json:"fqdn"`
IPAddresses []string `json:"ipAddresses"`
MACAddresses []string `json:"macAddresses"`
Owner string `json:"owner"`
OS *OS `json:"os"`
Software []Software `json:"software"`
Vulnerabilities []Vulnerability `json:"vulnerabilities"`
KICSRisks []*assets.KICSRisk `json:"kicsVulns"`
KSC *KSCFields `json:"ksc"`
Created string `json:"created"`
Updated string `json:"updated"`
}

type KSCFields struct {
NAgentID string `json:"nAgentID"`
KSCInstanceID string `json:"kscInstanceID"`
KSCMasterHostname string `json:"kscMasterHostname"`
LastVisible string `json:"lastVisible"`
}

type OS struct {
```

```
Name string `json:"name"`
Version uint64 `json:"version"`
}

type Software struct {
Name string `json:"name"`
Version string `json:"version"`
Vendor string `json:"vendor"`
}

type Vulnerability struct {
KasperskyID string `json:"kasperskyID"`
ProductName string `json:"productName"`
DescriptionUrl string `json:"descriptionUrl"`
RecommendedMajorPatch string `json:"recommendedMajorPatch"`
RecommendedMinorPatch string `json:"recommendedMinorPatch"`
SeverityStr string `json:"severityStr"`
Severity uint64 `json:"severity"`
CVE []string `json:"cve"`
ExploitExists bool `json:"exploitExists"`
MalwareExists bool `json:"malwareExists"`
}

type assets.KICSRisk struct {
ID int64 `json:"id"`
Name string `json:"name"`
Category string `json:"category"`
Description string `json:"description"`
DescriptionUrl string `json:"descriptionUrl"`
Severity int `json:"severity"`
Cvss float64 `json:"cvss"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Invalid value of the "page" parameter	invalid query parameter value	page
500	Any other internal errors	variable	variable

Importing assets

Details on identifying, creating, and updating assets

Assets are imported according to the asset data merging rules.

POST /api/v2/assets/import

Bulk creation or update of assets.

If the FQDN of an asset is specified, it acts as the unique ID of the asset within the tenant. If more than one FQDN is specified, the first FQDN from the specified array of FQDNs is used. If no FQDN is specified, the first IP address in the specified array of addresses is used to identify the asset. If the asset name is not specified, either FQDN or the first IP address is used as the name. Assets imported from KSC cannot be updated, therefore, FQDN conflicts may occur during the import process if a KSC asset with a the same FQDN already exists in the tenant. Such conflicts prevent the processing of the conflicting asset, but do not prevent the processing of other assets specified in the request body. Allows you to populate custom fields by uuid from the assetsCustomFields settings.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Request body

Format: JSON

```
type Request struct {
TenantID string `json:"tenantID"`
    Assets []Asset `json:"assets"`
}

type Asset struct {
Name string `json:"name"`
FQDN string `json:"fqdn"`
IPAddresses []string `json:"ipAddresses"`
MACAddresses []string `json:"macAddresses"`
Owner string `json:"owner"`
OS *OS `json:"os"`
Software []Software `json:"software"`
Vulnerabilities []Vulnerability `json:"vulnerabilities"`
CustomFields []CustomField `json:"customFields"`
}

type OS struct {
Name string `json:"name"`
Version uint64 `json:"version"`
}

type Software struct {
Name string `json:"name"`
Version string `json:"version"`
Vendor string `json:"vendor"`
}

type Vulnerability struct {
KasperskyID string `json:"kasperskyID"`
ProductName string `json:"productName"`
DescriptionUrl string `json:"descriptionUrl"`
RecommendedMajorPatch string `json:"recommendedMajorPatch"`
RecommendedMinorPatch string `json:"recommendedMinorPatch"`
SeverityStr string `json:"severityStr"`
Severity uint64 `json:"severity"`
CVE []string `json:"cve"`
ExploitExists bool `json:"exploitExists"`
MalwareExists bool `json:"malwareExists"`
}

type CustomFields struct {
```

```
ID string `json:"id"`
Value string `json:"value"`
}
```

Request mandatory fields

Name	Data type	Mandatory	Description	Value example
TenantID	string	Yes	Tenant ID	00000000-0000-0000-0000-000000000000
assets	[]Asset	Yes	Array of imported assets	

Asset mandatory fields

Name	Data type	Mandatory	Description	Value example
fqdn	string	If the ipAddresses array is not specified	Asset FQDN. You can specify multiple values separated by commas. It is recommended that you specify the FQDN and not just the host name. Priority indicator for asset identification.	[my-asset-1.example.com] [my-asset-1]
ipAddresses	[]string	If FQDN is not specified	Array of IP addresses for the asset. IPv4 or IPv6. The first element of the array is used as a secondary indicator for asset identification.	["192.168.1.1", "192.168.2.2"] ["2001:0db8:85a3:0000:0000:8a2e:0370:7334"]

Response

HTTP code: 200

Format: JSON

```
type Response struct {
InsertedIDs map[int64]interface{} `json:"insertedIDs"`
UpdatedCount uint64 `json:"updatedCount"`
Errors []ImportError `json:"errors"`
}
```

```
type ImportError struct {
Index uint64 `json:"index"`
Message string `json:"message"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Tenant ID is not specified	tenantID required	-
400	Attempt to import assets into the shared tenant	import into shared tenant not allowed	-
400	Not a single asset was specified in the request body	at least one asset required	-
400	None of the mandatory fields is specified	one of fields required	asset[<index>]: fqdn, ipAddresses
400	Invalid FQDN	invalid value	asset[<index>].fqdn
400	Invalid IP address	invalid value	asset[<index>].ipAddresses[<index>]
400	IP address is repeated	duplicated value	asset[<index>].ipAddresses
400	Invalid MAC address	invalid value	asset[<index>].macAddresses[<index>]
400	MAC address is repeated	duplicated value	asset[<index>].macAddresses
403	The user does not have the required role in the specified tenant	access denied	-
404	The specified tenant was not found	tenant not found	-
406	The specified tenant was disabled	tenant disabled	-
500	Any other internal errors	variable	variable

Deleting assets

POST /api/v2/assets/delete

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Request body

Format: JSON

Name	Data	Mandatory	Description	Value example

	type			
TenantID	string	Yes	Tenant ID	00000000-0000-0000-0000-000000000000
ids	[]string	If neither the ipAddresses array nor the FQDNs are specified	List of asset IDs	["00000000-0000-0000-0000-000000000000"]
fqdns	[]string	If neither the ipAddresses array nor the IDs are specified	Array of asset FQDNs	["my-asset-1.example.com", "my-asset-1"]
ipAddresses	[]string	If neither the IDs nor FQDNs are specified	Array of main IP addresses of the asset.	["192.168.1.1", "2001:0db8:85a3:0000:0000:8a2e:0370:7334"]

Response

HTTP code: 200

Format: JSON

```
type Response struct {
DeletedCount uint64 `json:"deletedCount"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Tenant ID is not specified	tenantID required	-
400	Attempt to delete an asset from the shared tenant	delete from shared tenant not allowed	-
400	None of the mandatory fields is specified	one of fields required	ids, fqdns, ipAddresses
400	Invalid FQDN specified	invalid value	fqdns[<index>]
400	Invalid IP address specified	invalid value	ipAddresses[<index>]
403	The user does not have the required role in the specified tenant	access denied	-
404	The specified tenant was not found	tenant not found	-
406	The specified tenant was disabled	tenant disabled	-
500	Any other internal errors	variable	variable

Searching events

POST /api/v2/events

Only search queries or aggregation queries (SELECT) are allowed.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, Access to NCIRCC, Access to CII.

Request body

Format: JSON

Request

Name	Data type	Mandatory	Description	Value example
period	Period	Yes	Search period	

sql	string	Yes	SQL query	SELECT * FROM events WHERE Type = 3 ORDER BY Timestamp DESC LIMIT 1000
				SELECT sum(BytesOut) as TotalBytesSent, SourceAddress FROM events WHERE DeviceVendor = 'netflow' GROUP BY SourceAddress LIMIT 1000
				SELECT count(Timestamp) as TotalEvents FROM events LIMIT 1

ClusterID	string	No, if the cluster is the only one	Storage cluster ID. You can find it by requesting a list of services with kind = storage. The cluster ID will be in the resourceID field.	00000000-0000-0000-0000-000000000000
rawTimestamps	bool	No	Display timestamps in their current format—Milliseconds since EPOCH. False by default.	true or false
emptyFields	bool	No	Display empty fields for normalized events. False by default.	true or false

Period

Name	Data type	Mandatory	Description	Value example
from	string	Yes	Lower bound of the period in RFC3339 format. Timestamp >= <from>	2021-09-06T00:00:00Z (UTC) 2021-09-06T00:00:00.000Z (UTC, including milliseconds) 2021-09-06T00:00:00Z+00:00 (MSK)
to	string	Yes	Upper bound of the period in RFC3339 format. Timestamp <= <to>	2021-09-06T00:00:00Z (UTC) 2021-09-06T00:00:00.000Z (UTC, including milliseconds)

				2021-09-06T00:00:00Z+00:00 (MSK)

Response

HTTP code: 200

Format: JSON

Result of executing the SQL query

Possible errors

HTTP code	Description	message field value	details field value
400	The lower bounds of the range is not specified	period.from required	-
400	The lower bounds of the range is in an unsupported format	cannot parse period.from	variable
400	The lower bounds of the range is equal to zero	period.from cannot be 0	-
400	The upper bounds of the range is not specified	period.to required	-
400	The upper bounds of the range is in an unsupported format	cannot parse period.to	variable
400	The upper bounds of the range is equal to zero	period.to cannot be 0	-
400	The lower bounds of the range is greater than the upper bounds	period.from cannot be greater than period.to	-
400	Invalid SQL query	invalid sql	variable
400	An invalid table appears in the SQL query	the only valid table is `events`	-
400	The SQL query lacks a LIMIT	sql: LIMIT required	-
400	The LIMIT in the SQL query exceeds the maximum (1000)	sql: maximum LIMIT is 1000	-
404	Storage cluster not found	cluster not found	-
406	The clusterID parameter was not specified, and many clusters were registered in KUMA	multiple clusters found, please provide clusterID	-
500	No available cluster nodes	no nodes available	-
50x	Any other internal errors	event search failed	variable

Viewing information about the cluster

GET /api/v2/events/clusters

Access: The main tenant clusters are accessible to all users.

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
page	number	No	Page number. Starts with 1. The page size is 250 entries. If the parameter is not specified, the default value is 1.	1
id	string	No	Cluster ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied	00000000-0000-0000-0000-000000000000
TenantID	string	No	Tenant ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied. If the user does not have the required role in the specified tenant, then this tenant is ignored.	00000000-0000-0000-0000-000000000000
name	string	No	Cluster name. Case-insensitive regular expression (PCRE).	cluster ^My cluster$

Response

HTTP code: 200

Format: JSON

```
type Response []Cluster

type Cluster struct {
ID string `json:"id"`
Name string `json:"name"`
TenantID string `json:"tenantID"`
TenantName string `json:"tenantName"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Invalid value of the "page" parameter	invalid query parameter value	page
500	Any other internal errors	variable	variable

Resource search

GET /api/v2/resources

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Access to shared resources.

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
page	number	No	Page number. Starts with 1. The page size is 250 entries. If the parameter is not specified, the default value is 1.	1
id	string	No	Resource ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied.	00000000-0000-0000-0000-000000000000
TenantID	string	No	Resource tenant ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied. If the user does not have the required role in the specified tenant, then this tenant is ignored.	00000000-0000-0000-0000-000000000000
name	string	No	Resource name. Case-insensitive regular	resource ^My resource$

			expression (PCRE).	
kind	string	No	Resource type. If the parameter is specified several times, then a list is generated and the logical OR operator is applied	collector, correlator, storage, activeList, aggregationRule, con enrichmentRule, destination, filter, normalizer, responseRule, s

Response

HTTP code: 200

Format: JSON

```
type Response []Resource

type Resource struct {
ID string `json:"id"`
Kind string `json:"kind"`
Name string `json:"name"`
Description string `json:"description"`
TenantID string `json:"tenantID"`
TenantName string `json:"tenantName"`
UserID string `json:"userID"`
UserName string `json:"userName"`
Created string `json:"created"`
Updated string `json:"updated"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Invalid value of the "page" parameter	invalid query parameter value	page
400	Invalid value of the "kind" parameter	invalid kind	<kind>
500	Any other internal errors	variable	variable

Loading resource file

POST /api/v2/resources/upload

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Request body

Encrypted contents of the resource file in binary format.

Response

HTTP code: 200

Format: JSON

File ID. It should be specified in the body of requests for viewing the contents of the file and for importing resources.

```
type Response struct {
ID string `json:"id"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	The file size exceeds the maximum allowable (64 MB)	maximum file size is 64 MB	-
403	The user does not have the required roles in any of the tenants	access denied	-
500	Any other internal errors	variable	variable

Viewing the contents of a resource file

POST /api/v2/resources/toc

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Request body

Format: JSON

Name	Data type	Mandatory	Description	Value example
fileID	string	Yes	The file ID obtained as a result of loading the resource file.	00000000-0000-0000-0000-000000000000
password	string	Yes	Resource file password.	SomePassword!88

Response

HTTP code: 200

Format: JSON

File version, list of resources, categories, and folders.

The ID of the retrieved resources must be used when importing.

```
type TOCResponse struct {
Folders []*Folder `json:"folders"`
}

type Folder struct {
ID string `json:"id"`
TenantID string `json:"tenantID"`
TenantName string `json:"tenantName"`
ExportID string `json:"exportID"`
Kind string `json:"kind"`
SubKind string `json:"subKind"`
Name string `json:"name"`
Description string `json:"description"`
UserID string `json:"userID"`
ParentID string `json:"parentID"`
CreatedAt int64 `json:"createdAt"`
Resources []*Resource `json:"resources"`
}

type Resource struct {
ID string `json:"id"`
Kind string `json:"kind"`
Name string `json:"name"`
Deps []string `json:"deps"`
}
```

Importing resources

POST /api/v2/resources/import

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Request body

Name	Data type	Mandatory	Description	Value example
fileID	string	Yes	The file ID obtained as a result of loading the resource file.	00000000-0000-0000-0000-000000000000
password	string	Yes	Resource file password.	SomePassword!88

TenantID	string	Yes	ID of the target tenant	00000000-0000-0000-0000-000000000000

actions	map[string]uint8	Yes	Mapping of the resource ID to the action that must be taken in relation to it.	0—do not import (used when resolving conflicts)
				1—import (should initially be assigned to each resource)
				2—replace (used when resolving conflicts)
				`{`
				`"00000000-0000-0000-0000-000000000000": 0,`
				`"00000000-0000-0000-0000-000000000001": 1,`
				`"00000000-0000-0000-0000-000000000002": 2,`
				`}`

Response

HTTP code	Body
204	

409	The imported resources conflict with the existing ones by ID. In this case, you need to repeat the import operation while specifying the following actions for these resources:
	0—do not import
	2—replace
	`type ImportConflictsError struct {`
	``HardConflicts []string `json:"conflicts"` ``
	`}`

Exporting resources

POST /api/v2/resources/export

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Access to shared resources.

Request body

Format: JSON

Name	Data type	Mandatory	Description	Value example
ids	[]string	Yes	Resource IDs to be exported	["00000000-0000-0000-0000-000000000000"]
password	string	Yes	Exported resource file password	SomePassword!88
TenantID	string	Yes	ID of the tenant that owns the exported resources	00000000-0000-0000-0000-000000000000

Response

HTTP code: 200

Format: JSON

ID of the file with the exported resources. It should be used in a request to download the resource file.

```
type ExportResponse struct {
FileID string `json:"fileID"`
}
```

Downloading the resource file

GET /api/v2/resources/download/<id>

Here "id" is the file ID obtained as a result of executing a resource export request.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Response

HTTP code: 200

Encrypted contents of the resource file in binary format.

Possible errors

HTTP code	Description	message field value	details field value
400	File ID not specified	route parameter required	id
400	The file ID is not a valid UUID	id is not a valid UUID	-

403	The user does not have the required roles in any of the tenants	access denied	-
404	File not found	file not found	-
406	The file is a directory	not regular file	-
500	Any other internal errors	variable	variable

Search for services

GET /api/v2/services

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
page	number	No	Page number. Starts with 1. The page size is 250 entries. If the parameter is not specified, the default value is 1.	1
id	string	No	Service ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied.	00000000-0000-0000-0000-000000000000
TenantID	string	No	Service tenant ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied. If the user does not have the required role in the specified tenant, then this tenant is ignored.	00000000-0000-0000-0000-000000000000
name	string	No	Service name. Case-insensitive regular expression (PCRE).	service ^My service$
kind	string	No	Service type. If the parameter is specified several times, then a list is generated and the logical OR operator is applied.	collector, correlator, storage, agent
fqdn	string	No	Service FQDN. Case-insensitive regular expression (PCRE).	hostname ^hostname.example.com$
paired	bool	No	Display only those services that executed the first start. If the parameter is present in the URL query, then its value is assumed to be true. The values specified by the user are ignored. Example: /api/v1/services?paired	

Response

HTTP code: 200

Format: JSON

```
type Response []Service

type Service struct {
ID string `json:"id"`
TenantID string `json:"tenantID"`
TenantName string `json:"tenantName"`
ResourceID string `json:"resourceID"`
Kind string `json:"kind"`
Name string `json:"name"`
Address string `json:"address"`
FQDN string `json:"fqdn"`
Status string `json:"status"`
Warning string `json:"warning"`
APIPort string `json:"apiPort"`
Uptime string `json:"uptime"`
Version string `json:"version"`
Created string `json:"created"`
Updated string `json:"updated"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Invalid value of the "page" parameter	invalid query parameter value	page
400	Invalid value of the "kind" parameter	invalid kind	<kind>
500	Any other internal errors	variable	variable

Tenant search

GET /api/v2/tenants

Only tenants available to the user are displayed.

Access: General administrator, Administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, Interaction with NCIRCC, Access to CII, Access to shared resources.

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example

page	number	No	Page number. Starts with 1. The page size is 250 entries. If the parameter is not specified, the default value is 1.	1
id	string	No	Tenant ID. If the parameter is specified several times, then a list is generated and the logical OR operator is applied.	00000000-0000-0000-0000-000000000000
name	string	No	Tenant name. Case-insensitive regular expression (PCRE).	tenant ^My tenant$
main	bool	No	Only display the main tenant. If the parameter is present in the URL query, then its value is assumed to be true. The values specified by the user are ignored. Example: /api/v1/tenants?main	

Response

HTTP code: 200

Format: JSON

```
type Response []Tenant

type Tenant struct {
    ID          string `json:"id"`
    Name        string `json:"name"`
    Main        bool   `json:"main"`
    Description string `json:"description"`
    EPS         uint64 `json:"eps"`
    EPSLimit    uint64 `json:"epsLimit"`
    Created     string `json:"created"`
    Updated     string `json:"updated"`
Shared   bool   `json:"shared"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Invalid value of the "page" parameter	invalid query parameter value	page
500	Any other internal errors	variable	variable

View token bearer information

GET /api/v2/users/whoami

Response

HTTP code: 200

Format: JSON

```
type Tenant struct {
ID string `json:"id"`
Name string `json:"name"`
}

type Role struct {
ID string `json:"id"`
Name string `json:"name"`
Tenants []Tenant `json:"tenants"`
}

type Response struct {
ID string `json:"id"`
Name string `json:"name"`
Login string `json:"login"`
Email string `json:"email"`
Roles []Role `json:"roles"`
}
```

Dictionary updating in services

POST /api/v2/dictionaries/update

You can update only dictionaries in dictionary resources of the table type.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
dictionaryID	string	Yes	ID of the dictionary that will be updated.	00000000-0000-0000-0000-000000000000

The update affects all services where the specified dictionary is used. If an update in one of the services ends with an error, this does not interrupt updates in the other services.

Request body

Multipart field name	Data type	Mandatory	Description	Value example
file	CSV file	Yes	The request contains a CSV file. Data of the existing dictionary is being replaced with data from this file. The first line of the CSV file containing the column names must not be changed.	key columns,column1,column2 key1,k1col1,k1col2 key2,k2col1,k2col2

Response

HTTP code: 200

Format: JSON

```
type Response struct {
ServicesFailedToUpdate []UpdateError `json:"servicesFailedToUpdate"`
}
type UpdateError struct {
ID string `json:"id"`
Err error `json:"err"`
}
```

Returns only errors for services in which the dictionaries have not been updated.

Possible errors

HTTP code	Description	message field value	details field value
400	Invalid request body	request body decode failed	Error
400	Null count of dictionary lines	request body required	-
400	Dictionary ID not specified	invalid value	dictionaryID
400	Incorrect value of dictionary line	invalid value	rows or rows[i]
400	Dictionary with the specified ID has an invalid type (not table)	can only update table dictionary	-
400	Attempt to change dictionary columns	columns must not change with update	-
403	No access to requested resource	access denied	-
404	Service not found	service not found	-
404	Dictionary not found	dictionary not found	Service ID
500	Any other internal errors	variable	variable

Dictionary retrieval

GET /api/v2/dictionaries

You can get only dictionaries in dictionary resources of the table type.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
dictionaryID	string	Yes	ID of the dictionary that will be received	00000000-0000-0000-0000-000000000000

Response

HTTP code: 200

Format: text/plain; charset=utf-8

A CSV file is returned with the dictionary data in the response body.

Viewing custom fields of the assets

GET /api/v2/settings/id/:id

The user can view a list of custom fields made by the KUMA user in the application web interface.

A custom field is a bucket for entering text. If necessary, the default value and the mask can be used to validate the entered text in the following format: https://pkg.go.dev/regexp/syntax. All forward slash characters in the mask must be shielded.

Access: General administrator, Main tenant administrator.

Query parameters

Name	Data type	Mandatory	Description	Value example
id	string	Yes	Configuration ID of the custom fields	00000000-0000-0000-0000-000000000000

Response

HTTP code: 200

Format: JSON

```
type Settings struct {
ID string `json:"id"`
TenantID string `json:"tenantID"`
TenantName string `json:"tenantName"`
Kind string `json:"kind"`
UpdatedAt int64 `json:"updatedAt"`
CreatedAt int64 `json:"createdAt"`
Disabled bool `json:"disabled"`
CustomFields []*CustomField `json:"customFields"`
}
```

```
type CustomField struct {
ID string `json:"id"`
Name string `json:"name"`
Default string `json:"default"`
Mask string `json:"mask"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
404	Parameters not found: invalid ID or parameters are missing	Not found in database	null
500	Any other internal errors	variable	variable

Creating a backup of the KUMA Core

GET /api/v2/system/backup

Access: General administrator.

The request has no parameters.

The tar.gz archive containing the backup copy of the KUMA Core is returned in response to the request. The backup copy is not saved on the host where the Core is installed. The certificates are included in the backup copy.

If the operation is successful, an audit event is generated with the following parameters:

- `DeviceAction = "Core backup created"`
- `SourceUserID = "<user-login>"`

You can restore the KUMA Cores from a backup using the following API request: `POST /api/v2/system/restore`.

Restoring the KUMA Core from the backup

POST /api/v2/system/restore

Access: General administrator.

The request has no parameters.

The request body must contain an archive with the backup copy of the KUMA Core, obtained as a result of the following API request execution: `GET /api/v2/system/backup`.

After receiving the archive with the backup copy, KUMA performs the following actions:

1. Unpacks the archive with the backup copy of the KUMA Core to a temporary directory.
2. Compares the current KUMA version with the backup KUMA version. Data may only be restored from a backup if it is restored to the KUMA of the same version as the backup one.

   If the versions match, an audit event is generated with the following parameters:

   - `DeviceAction = "Core restore scheduled"`
   - `SourceUserID = "<name of the user who initiated KUMA restore from a backup copy>"`
3. If the versions match, data is restored from the backup copy of the KUMA Core.
4. The temporary directory is deleted, and KUMA starts normally.

   The "WARN: restored from backup" entry is added to the KUMA Core log.

Viewing the list of context tables in the correlator

GET /api/v2/contextTables

The target correlator must be running.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
correlatorID	string	Yes	Correlator service ID	00000000-0000-0000-0000-000000000000

Response

HTTP code: 200

Format: JSON

```
type Response []ContextTableInfo

type ContextTableInfo struct {
ID string `json:"id"`
Name string `json:"name"`
Dir string `json:"dir"`
Records uint64 `json:"records"`
WALSize uint64 `json:"walSize"`
}
```

Possible errors

HTTP code	Description	message field value	details field value
400	Correlator service ID is not specified.	query parameter required	correlatorID
403	The user does not have the required role in the correlator tenant.	access denied	-
404	The service with the specified ID (correlatorID) was not found.	service not found	-
406	The service with the specified ID (correlatorID) is not a correlator.	service is not correlator	-
406	The correlator did not execute the first start.	service not paired	-
406	The tenant of the correlator is disabled.	tenant disabled	-
50x	Failed to gain access to the correlator API.	correlator API request failed	variable
500	Failed to decode the body of the response received from the correlator.	correlator response decode failed	variable
500	Any other internal error.	variable	variable

Importing records into a context table

POST /api/v2/contextTables/import

The target correlator must be running.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst (can import data into any correlator table of an accessible tenant, even if the context table was created in the Shared tenant).

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
correlatorID	string	Yes	Correlator service ID	00000000-0000-0000-0000-000000000000
contextTableID	string	If contextTableName is not specified	Context table ID	00000000-0000-0000-0000-000000000000
contextTableName	string	If contextTableID is not specified	Name of the context table	Attackers
format	string	Yes	Format of imported entries	CSV, TSV, internal
clear	bool	No	Clear the context table before importing. If the	/api/v2/contextTables/import?clear

			parameter is present in the URL query, its value is assumed to be true. The values specified by the user are ignored.	

Request body

Format	Contents
CSV	The first row is the header, which lists the comma-separated fields. The rest of the rows are the comma-separated values corresponding to the fields in the header. The number of fields in each row must be the same, and it must match the number of fields in the schema of the context table. List field values are separated by the "|" character. For example, the value of a list of integers might be 1|2|3.
TSV	The first row is the header, which lists the TAB-separated fields. The rest of the rows are the TAB-separated values corresponding to the fields in the header. The number of fields in each row must be the same, and it must match the number of fields in the schema of the context table. List field values are separated by the "|" character.
internal	Each line contains one individual JSON object. Data in the 'internal' format can be obtained by exporting the contents of the context table from the correlator in the KUMA web console.

Response

HTTP code: 204

Possible errors

HTTP code	Description	message field value	details field value
400	Correlator service ID is not specified.	query parameter required	correlatorID
400	Neither the contextTableID parameter nor the contextTableName parameter is specified	one of query parameters required	contextTableID, contextTableName
400	The 'format' parameter is not specified	query parameter required	format
400	The 'format' parameter is invalid	invalid query parameter value	format
400	The request body has zero length	request body required	-
400	Error parsing the request body, including the non-conformance of the field names and types of the record being imported with the schema of the context table.	correlator API request failed	variable
403	The user does not have the required role in the correlator tenant.	access denied	-
404	The service with the specified ID (correlatorID) was not	service not	-

	found.	found	
404	The context table was not found.	context table not found	–
406	The service with the specified ID (correlatorID) is not a correlator.	service is not correlator	–
406	The correlator did not execute the first start.	service not paired	–
406	The tenant of the correlator is disabled.	tenant disabled	–
406	More than one context table found by a search for contextTableName.	more than one matching context tables found	–
50x	Failed to gain access to the correlator API.	correlator API request failed	variable
500	Error preparing data for importing into the correlator service.	context table process import request failed	variable
500	Any other internal error.	variable	variable

Exporting records from a context table

GET /api/v2/contextTables/export

The target correlator must be running.

Access: General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst.

Query parameters (URL Query)

Name	Data type	Mandatory	Description	Value example
correlatorID	string	Yes	Correlator service ID	00000000-0000-0000-0000-000000000000
contextTableID	string	If contextTableName is not specified	Context table ID	00000000-0000-0000-0000-000000000000
contextTableName	string	If contextTableID is not specified	Name of the context table	Attackers

Response

HTTP code: 200

Format: application/octet-stream

Body: exported context table data, in the 'internal' format: each row contains one individual JSON object.

Possible errors

HTTP code	Description	message field value	details field value
400	Correlator service ID is not specified.	query parameter required	correlatorID
400	Neither the contextTableID parameter nor the contextTableName parameter is specified	one of query parameters required	contextTableID, contextTableName
403	The user does not have the required role in the correlator tenant.	access denied	-
404	The service with the specified ID (correlatorID) was not found.	service not found	-
404	The context table was not found.	context table not found	-
406	The service with the specified ID (correlatorID) is not a correlator.	service is not correlator	-
406	The correlator did not execute the first start.	service not paired	-
406	The tenant of the correlator is disabled.	tenant disabled	-
406	More than one context table found by a search for contextTableName.	more than one matching context tables found	-
50x	Failed to gain access to the correlator API.	correlator API request failed	variable
500	Any other internal error.	variable	variable

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 version of the program.

Flags:

`-h`, `--h` are used to get help about any kuma command. For example, `kuma <component> --help`.

Examples:

- `kuma version` is used to get version of the KUMA installer.
- `kuma core -h` is used to get help about `core` command of KUMA installer.
- `kuma collector --core <address of the server where the collector should obtain its settings> --id <ID of the installed service> --api.port <port>` is used to start collector service installation.

Integrity check of KUMA files

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 in the/opt/kaspersky/kuma/integrity/manifest/* directory, signed with 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 screens 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 pick the command that matches the KUMA component you want to check:
   - `./check_all.sh` for KUMA Core and Storage components.
   - `./check_core.sh` for KUMA Core components.
   - `./check_collector.sh` for KUMA collector components.
   - `./check_collector.sh` for KUMA correlator components.
   - `./check_storage.sh` for storage components.
   - `./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\.

   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.

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	
The name of a field reflects its purpose. Th			
ApplicationProtocol	String	31 characters	Name of the application layer
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 as xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:x
DestinationCity	String	1,023 characters	City corresponding to the IP a
DestinationCountry	String	1023 characters	Country corresponding to the
DestinationDnsDomain	String	255 characters	The DNS portion of the fully c
DestinationHostName	String	1023 characters	Host name of the destination.
DestinationLatitude	Floating point number	+/- 1.7E-308 to 1.7E+308	Longitude corresponding to t
DestinationLongitude	Floating point number	+/- 1.7E-308 to 1.7E+308	Latitude corresponding to the
DestinationMacAddress	String	17 characters	MAC address of the destinati
DestinationNtDomain	String	255 characters	Windows Domain Name of the
DestinationPort	Number	From -9223372036854775808 to 9223372036854775807	Port number of the destinatio
DestinationProcessID	Number	From -9223372036854775808 to 9223372036854775807	System process ID registered
DestinationProcessName	String	1023 characters	Name of the system process r
DestinationRegion	String	1023 characters	Region corresponding to the I
DestinationServiceName	String	1023 characters	Name of the service on the de
DestinationTranslatedAddress	String	45 characters	Translated IPv4 or IPv6 addres
DestinationTranslatedPort	Number	From -9223372036854775808 to 9223372036854775807	Port number at the destinatio
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 u
DeviceAction	String	63 characters	Action that was taken by the
DeviceAddress	String	45 characters	IPv4 or IPv6 address of the de xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:x
DeviceCity	String	1023 characters	City corresponding to the IP
DeviceCountry	String	1023 characters	Country corresponding to the
DeviceDnsDomain	String	255 characters	DNS part of the fully qualified
DeviceEventClassID	String	1023 characters	Event type ID assigned by the
DeviceExternalID	String	255 characters	ID of the device or product as
DeviceFacility	String	1023 characters	Value of the facility paramete
DeviceHostName	String	100 characters	Name of the device from whic
DeviceInboundinterface	String	128 characters	Name of the incoming connec
DeviceLatitude	Floating point number	+/- 1.7E-308 to 1.7E+308	Longitude corresponding to t
DeviceLongitude	Floating point number	+/- 1.7E-308 to 1.7E+308	Latitude corresponding to the
DeviceMacAddress	String	17 characters	MAC address of the asset fro
DeviceNtDomain	String	255 characters	Windows Domain Name of the
DeviceOutboundinterface	String	128 characters	Name of the outgoing connec
DevicePayloadID	String	128 characters	The payload's unique ID that is
DeviceProcessID	Number	From -9223372036854775808 to 9223372036854775807	ID of the system process on t
DeviceProcessName	String	1023 characters	Name of the process.
DeviceProduct	String	63 characters	Name of the product that ger the log source.
DeviceReceiptTime	Number	From -9223372036854775808 to 9223372036854775807	Time when the device receive
DeviceRegion	String	1023 characters	Region corresponding to the I
DeviceTimeZone	String	255 characters	Time zone of the device on wh
DeviceTranslatedAddress	String	45 characters	Re-translated IPv4 or IPv6 add xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:x
DeviceVendor	String	63 characters	Vendor name of the event so source.
DeviceVersion	String	31 characters	Product version of the event source.
EndTime	Number	From -9223372036854775808	Date and time (timestamp) wh

		to 9223372036854775807	
EventOutcome	String	63 characters	Result of the operation. For e
ExternalID	String	40 characters	Field in which the ID can be sa
FileCreateTime	Number	From -9223372036854775808 to 9223372036854775807	File creation time.
FileHash	String	255 characters	Hash of the file. Example: CA737F1014A48F4C0B6DD4
FileID	String	1023 characters	ID of the file.
FileModificationTime	Number	From -9223372036854775808 to 9223372036854775807	Time when the file was last m
FileName	String	1023 characters	Filename without specifying t
FilePath	String	1023 characters	File path, including the file nam
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 c displayed based in the timezo
OldFileHash	String	255 characters	Hash of the OLD file. Example CA737F1014A48F4C0B6DD4
OldFileID	String	1023 characters	ID of the OLD file.
OldFileModificationTime	Number	From -9223372036854775808 to 9223372036854775807	Time when the OLD file was la
OldFileName	String	1023 characters	Name of the OLD file (without
OldFilePath	String	1023 characters	Path to the OLD file, including
OldFilePermission	String	1023 characters	List of permissions of the OL
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

RequestClientApplication	String	1023 characters	Value of the "user-agent" para
RequestContext	String	2,048 characters	Description of the http reque
RequestCookies	String	1023 characters	Cookies associated with the h
RequestMethod	String	1023 characters	Method used when making th
RequestUrl	String	1023 characters	Requested URL.
Severity	String	1023 characters	Priority. This can be the Sever
SourceAddress	String	45 characters	IPv4 or IPv6 address of the sc
SourceCity	String	1023 characters	City corresponding to the IP a
SourceCountry	String	1023 characters	Country corresponding to the
SourceDnsDomain	String	255 characters	The DNS portion of the fully c
SourceHostName	String	1023 characters	Windows Domain Name of the
SourceLatitude	Floating point number	+/- 1.7E-308 to 1.7E+308	Longitude corresponding to t
SourceLongitude	Floating point number	+/- 1.7E-308 to 1.7E+308	Latitude corresponding to the
SourceMacAddress	String	17 characters	MAC address of the source. F
SourceNtDomain	String	255 characters	Windows Domain Name of the
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 a
SourceRegion	String	1023 characters	Region corresponding to the I
SourceServiceName	String	1023 characters	Name of the service on the sc
SourceTranslatedAddress	String	45 characters	Translated IPv4 or IPv6 addres
SourceTranslatedPort	Number	From -9223372036854775808 to 9223372036854775807	Port number of the source aft
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 u
StartTime	Number	From -9223372036854775808 to 9223372036854775807	Date and time (timestamp) wh
Tactic	String	128 characters	Name of the tactic from the M
Technique	String	128 characters	Name of the technique from t

TransportProtocol	String	31 characters	Name of the Transport layer p
Type	Number	From -9223372036854775808 to 9223372036854775807	Event type: 1 - basic, 2 - aggre
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 t displayed based in the timezo
DeviceCustomDate1Label	String	1023 characters	Field for describing the purpo
DeviceCustomDate2	Number, timestamp	From -9223372036854775808 to 9223372036854775807	Field for mapping a date and t displayed based in the timezo
DeviceCustomDate2Label	String	1023 characters	Field for describing the purpo
DeviceCustomFloatingPoint1	Floating point number	+/- 1.7E-308 to 1.7E+308	Field for mapping floating poi
DeviceCustomFloatingPoint1Label	String	1023 characters	Field for describing the purpo
DeviceCustomFloatingPoint2	Floating point number	+/- 1.7E-308 to 1.7E+308	Field for mapping floating poi
DeviceCustomFloatingPoint2Label	String	1023 characters	Field for describing the purpo
DeviceCustomFloatingPoint3	Floating point number	+/- 1.7E-308 to 1.7E+308	Field for mapping floating poi
DeviceCustomFloatingPoint3Label	String	1023 characters	Field for describing the purpo
DeviceCustomFloatingPoint4	Floating point number	+/- 1.7E-308 to 1.7E+308	Field for mapping floating poi
DeviceCustomFloatingPoint4Label	String	1023 characters	Field for describing the purpo
DeviceCustomIPv6Address1	String	45 characters	Field for mapping an IPv6 add
DeviceCustomIPv6Address1Label	String	1023 characters	Field for describing the purpo
DeviceCustomIPv6Address2	String	45 characters	Field for mapping an IPv6 add
DeviceCustomIPv6Address2Label	String	1023 characters	Field for describing the purpo
DeviceCustomIPv6Address3	String	45 characters	Field for mapping an IPv6 add
DeviceCustomIPv6Address3Label	String	1023 characters	Field for describing the purpo
DeviceCustomIPv6Address4	String	45 characters	Field for mapping an IPv6 add
DeviceCustomIPv6Address4Label	String	1023 characters	Field for describing the purpo
DeviceCustomNumber1	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer v
DeviceCustomNumber1Label	String	1023 characters	Field for describing the purpo
DeviceCustomNumber2	Number	From -9223372036854775808	Field for mapping an integer v

		to 9223372036854775807	
DeviceCustomNumber2Label	String	1023 characters	Field for describing the purpo
DeviceCustomNumber3	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer v
DeviceCustomNumber3Label	String	1023 characters	Field for describing the purpo
DeviceCustomString1	String	4,000 characters	Field for mapping a string valu
DeviceCustomString1Label	String	1,023 characters	Field for describing the purpo
DeviceCustomString2	String	4,000 characters	Field for mapping a string valu
DeviceCustomString2Label	String	1023 characters	Field for describing the purpo
DeviceCustomString3	String	4,000 characters	Field for mapping a string valu
DeviceCustomString3Label	String	1023 characters	Field for describing the purpo
DeviceCustomString4	String	4,000 characters	Field for mapping a string valu
DeviceCustomString4Label	String	1023 characters	Field for describing the purpo
DeviceCustomString5	String	4,000 characters	Field for mapping a string valu
DeviceCustomString5Label	String	1023 characters	Field for describing the purpo
DeviceCustomString6	String	4,000 characters	Field for mapping a string valu
DeviceCustomString6Label	String	1023 characters	Field for describing the purpo
DeviceDirection	Number	From -9223372036854775808 to 9223372036854775807	Field for describing the direct
DeviceEventCategory	String	1023 characters	Event category assigned by tl
FlexDate1	Number, timestamp	From -9223372036854775808 to 9223372036854775807	Field for mapping a date and t displayed based in the timezo
FlexDate1Label	String	128 characters	Field for describing the purpo
FlexNumber1	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer v
FlexNumber1Label	String	128 characters	Field for describing the purpo
FlexNumber2	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer v
FlexNumber2Label	String	128 characters	Field for describing the purpo
FlexString1	String	1023 characters	Field for mapping a string valu
FlexString1Label	String	128 characters	Field for describing the purpo
FlexString2	String	1023 characters	Field for mapping a string valu

FlexString2Label	String	128 characters	Field for describing the purpo
Service fields. Cannot be edited.			
AffectedAssets	Nested [Affected] structure	-	Nested structure from which appear in alert events.
AggregationRuleID	String	-	ID of the aggregation rule.
AggregationRuleName	String	-	Name of the aggregation rule
BaseEventCount	Number	-	For an aggregated base event correlation event, this is the n correlation event.
BaseEvents	Nested [Event] list	-	Nested structure containing a
Code	String	-	In a base event, this is the coc
CorrelationRuleID	String	-	ID of the correlation rule.
CorrelationRuleName	String	-	Name of the correlation rule t
DestinationAccountID	String	-	This field stores the user ID.
DestinationAssetID	String	-	This field stores the asset ID c
DeviceAssetID	String	-	This field stores the ID of the
Extra	Nested [string:string] dictionary	-	During normalization of a raw fields. This field can be filled ir
GroupedBy	String	-	List of names of the fields tha
ID	String	-	Unique event ID of UUID type. correlator generates the ID of
Raw	String	-	Non-normalized text of the or
ReplayID	String	-	ID of the retroscan that gener
ServiceID	String	-	ID of the service instance: cor
ServiceName	String	-	Name of the microservice inst
SourceAccountID	String	-	This field stores the user ID.
SourceAssetID	String	-	This field stores the asset ID c
SpaceID	String	-	ID of the space.
TenantID	String	-	This field stores the ID of the
TI	Nested [string:string] dictionary	-	Field that contains categories indicators from an event.
TICategories	map[string]	-	This field contains categories
Timestamp	Number	-	Timestamp of the base event time is specified in UTC0. In th

Nested `Affected` structure

Field	Data type	Description
`Assets`	Nested `[AffectedRecord]` list	List and number of assets associated with the alert.
`Accounts`	Nested `[AffectedRecord]` list	List and number of user accounts associated with the alert.

Nested `AffectedRecord` structure

Field	Data type	Description
`Value`	String	ID of the asset or user account.
`Count`	Number	The number of times an asset or user account appears in alert-related events.

Fields generated by KUMA

KUMA generates the following fields that cannot be modified: BranchID, BranchName, DestinationAccountName, DestinationAssetName, DeviceAssetName, SourceAccountName, SourceAssetName, TenantName.

Configuring the data model of a normalized event from KATA EDR

To investigate the information, the IDs of the event and the KATA/EDR process must go to certain fields of the normalized event. To build a process tree for events coming from KATA/EDR, you must configure the copying of data from the fields of the raw events to the fields of the normalized event in KUMA normalizers as follows:

1. For any KATA/EDR events, you must configure normalization with copying of the following fields:
   - The `EventType` field of the KATA/EDR event must be copied to the `DeviceEventCategory` field of the normalized KUMA event.
   - The `HostName` field of the KATA/EDR event must be copied to the `DeviceHostName` field of the normalized KUMA event.
2. For any event where `DeviceProduct = 'KATA'`, normalization must be configured in accordance with the table below.

   Normalization of event fields from KATA/EDR

KATA/EDR event field	Normalized event field
IOATag	DeviceCustomIPv6Address2
	IOATag
IOAImportance	DeviceCustomIPv6Address1
	IOAImportance
FilePath	FilePath
FileName	FileName
Md5	FileHash
FileSize	FileSize

3. For events listed in the table below, additional normalization with field copying must be configured in accordance with the table.

Additional normalization with copying of event fields from KATA/EDR

Event	Raw event field	Normalized event field
Process	UniqueParentPid	FlexString1
	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName
AppLock	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName
BlockedDocument	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName
Module	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName
FileChange	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName
Driver	HostName	DeviceHostName
	FileName	FileName
	ProductName	DeviceCustomString5,
		ProductName
	ProductVendor	DeviceCustomString6
		ProductVendor
Connection	UniquePid	FlexString2
	HostName	DeviceHostName
	URI	RequestURL
	RemoteIP	DestinationAddress
	RemotePort	DestinationPort
PortListen	UniquePid	FlexString2
	HostName	DeviceHostName
	LocalIP	SourceAddress
	LocalPort	SourcePort
Registry	UniquePid	FlexString2
	HostName	DeviceHostName

	ValueName	DeviceCustomString5
		New Value Name
	KeyName	DeviceCustomString4
		New Key Name
	PreviousKeyName	FlexString2
		Old Key Name
	ValueData	DeviceCustomString6
		New Value Data
	PreviousValueData	FlexString1
		Old Value Data
	ValueType	FlexNumber1
		Value Type
	PreviousValueType	FlexNumber2
		Previous Value Type
SystemEventLog	UniquePid	FlexString2
	HostName	DeviceHostName
	OperationResult	EventOutcome
	EventId	DeviceCustomNumber3
		EventId
	EventRecordId	DeviceCustomNumber2
		EventRecordId
	Channel	DeviceCustomString6
		Channel
	ProviderName	SourceUserID
ThreatDetect	UniquePid	FlexString2
	HostName	DeviceHostName
	VerdictName	EventOutcome
	DetectedObjectType	OldFileType
	isSilent	FlexString1
		Is Silent
	RecordId	DeviceCustomString5
		Record ID
	DatabaseTimestamp	DeviceCustomDate2
		Database Timestamp
ThreatDetectProcessingResult	UniquePid	FlexString2
	HostName	DeviceHostName

	ThreatStatus	DeviceCustomString5
		Threat Status
PROCESS_INTERPRET_FILE_RUN	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName
	InterpretedFilePath	OldFilePath
	InterpretedFileSize	OldFileSize
	InterpretedFileHash	OldFileHash
PROCESS_CONSOLE_INTERACTIVE_INPUT	UniquePid	FlexString2
	HostName	DeviceHostName
	InteractiveInputText	DeviceCustomString4
		Command Line
AMSI SCAN	UniquePid	FlexString2
	HostName	DeviceHostName
	ObjectContent	DeviceCustomString5
		Object Content

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.
`Overflow`	`TRUE/FALSE` string	Indicator that the alert is overflowed, which means that the size of the alert and the events associated with it exceeds 16 MB. Possible values: • `true` • `false`

`MaxAssetsWeightStr`	String	Maximum severity of the asset categories associated with the alert.
`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 for Networks server IP address.
`KICSConnectorID`	Number	KICS for Networks connector ID.
`KICSDeviceID`	Number	KICS for Networks asset ID.
`KICSStatus`	String	KICS for Networks asset status.
`KICSHardware`	Nested `[KICSSystemInfo]` structure	Asset hardware information received from KICS for Networks.
`KICSSoft`	Nested `[KICSSystemInfo]` structure	Asset software information received from KICS for Networks.
`KICSRisks`	Nested `[KICSRisk]` structure	Asset vulnerability information received from KICS for Networks.
`Sources`	Nested `[Sources]` structure	Basic information about the asset from various sources.
`FromKSC`	String with `TRUE/FALSE` values	Indicator that asset details have been imported from 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]`	Asset software information received from KSC.

	structure	
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 for Networks risk ID.
`Name`	String	Risk name.
`Category`	String	Risk type.
`Description`	String	Risk description.
`DescriptionURL`	String	Link to risk description.
`Severity`	Number	Risk severity.
`Cvss`	Number	CVSS score.

Nested Sources structure

Field	Value type	Description
KSC	Nested [SourceInfo] structure	Asset information received from 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 for Networks.

Nested Sources structure

Field	Value type	Description
MACAddress	Nested list of strings	Asset MAC addresses.
IPAddressInt	Nested list of numbers	IP address in number format.
Owner	Nested [OwnerInfo] structure	Asset owner information.
OS	Nested [OS] structure	Asset operating system information.
displayName	String	Asset name.
IPAddress	Nested list of strings	Asset IP addresses.
FQDN	String	Asset FQDN.
Weight	Number	Asset importance.
Deleted	String with TRUE/FALSE values	Indicator of whether the asset has been marked for deletion from KUMA.
UpdatedAt	Number	Date of last update of the asset.

Nested 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.
	String	Tenant ID.

TenantID		
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. This section covers the KUMA audit events.

Event fields with general information

Every audit event has the event fields described below.

Event field name	Field value
ID	Unique event ID in the form of an UUID.
Timestamp	Event time.
DeviceHostName	The event source host. For audit events, it is the hostname where kuma-core is installed, because it 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. For audit event the value is `4`.
TenantID	ID of the main tenant.
DeviceVendor	`Kaspersky`
DeviceProduct	`KUMA`
EndTime	Event creation time.

User was successfully signed in or failed to sign in

Event field name	Field value
DeviceAction	`user login`
EventOutcome	`succeeded` or `failed`—the status depends on the success or failure of the operation.
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.
SourceUserID	User ID.
Message	Description of the error; appears only if an error occurred during login. Otherwise, the field will be empty.

User login successfully 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, 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 data.
SourceUserID	User ID that was used to change data.
DestinationUserName	User login whose data was changed.
DestinationUserID	User ID 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`

User role was successfully changed

Event field name	Field value
DeviceAction	`user role 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 data.

SourceUserID	User ID that was used to change data.
DestinationUserName	User login whose data was changed.
DestinationUserID	User ID whose data was changed.
DeviceCustomString1	Current value of the role.
DeviceCustomString1Label	`new role`
DeviceCustomString2	Value of the role before it was changed.
DeviceCustomString2Label	`old role`

Other data of the user was successfully 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, 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 data.
SourceUserID	User ID that was used to change data.
DestinationUserName	User login whose data was changed.
DestinationUserID	User ID whose data was changed.

User successfully 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.

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, 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.
SourceUserID	User ID.

User password was successfully 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, 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 data.
SourceUserID	User ID that was used to change data.
DestinationUserName	User login whose data was changed.
DestinationUserID	User ID whose data was changed.

User was successfully 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, 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 user account.
SourceUserID	User ID that was used to create 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 successfully 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, 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 for whom the data changes were made.
SourceUserID	ID of the user for whom the data changes were made.
DestinationUserPrivileges	Role name. Available values: general admin, admin, analyst, operator.
DeviceCustomString5	ID of the tenant used to assign the role.
DeviceCustomString5Label	`tenant ID`
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	`tenant name`

User role was successfully 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, 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 makes the changes.
SourceUserID	ID of the user who makes the changes.
DestinationUserName	Login of the user for whom the changes are made.
DestinationUserID	ID of the user for whom the changes are made.
DestinationUserPrivileges	Role name. Available values: general admin, admin, analyst, operator.
DeviceCustomString5	ID of the tenant used to assign the role.
DeviceCustomString5Label	`tenant ID`

DeviceCustomString6	Tenant name.
DeviceCustomString6Label	`tenant name`

User access token was successfully 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, 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 data.
SourceUserID	User ID that was used to change the data.
DestinationUserName	User login whose data was changed.
DestinationUserID	ID of the user whose data was changed.

Service was successfully 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, 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.
DeviceProcessName	Service name.
DeviceFacility	Service type.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	`tenant ID`

DeviceCustomString6	Tenant name.
DeviceCustomString6Label	`tenant name`

Service was successfully 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, 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 delete the service.
SourceUserID	User ID that was used to delete 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 successfully 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, 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 reset the service.
SourceUserID	User ID that was used to restart 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 successfully 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, 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 restart the service.
SourceUserID	User ID that was used to restart 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 successfully 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	Address that reported information about service start. It may be a proxy address if the information passed through a proxy.
SourcePort	Port that reported information about service start. It may be a proxy port if the information passed through a proxy.
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 successfully 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	Address that sent a service pairing request. It may be a proxy address if the request passed through a proxy.
SourcePort	Port that sent a service pairing request. It may be a proxy port if the request passed through a proxy.
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`

Storage 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, 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 delete partition.
SourceUserID	User ID that was used to delete partition.
Name	Index name.
Message	`deleted by user`

Storage partition was deleted automatically due to expiration

Event field name	Field value
DeviceAction	`partition deleted`
EventOutcome	`succeeded`

Name	Index name
SourceServiceName	`scheduler`
Message	`deleted by retention period settings`

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.

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.

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, 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 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.

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, 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 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.

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, 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 perform the import.
SourceUserID	User ID that was used to perform the import.
DeviceExternalID	Service ID for which an import was performed.
ExternalID	Active list ID.
Name	Active list name.
Message	If EventOutcome = `failed`, an error message can be found here.
DeviceCustomString5	Service tenant ID. Some errors prevent adding tenant information to the event.
DeviceCustomString5Label	`tenant ID`
DeviceCustomString6	Tenant name
DeviceCustomString6Label	`tenant name`

Active list was exported successfully

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.

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, 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 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`

Resource was successfully 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, 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 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 successfully 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, 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 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 successfully 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, 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 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`

Asset was successfully 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, 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 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 successfully 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, 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 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 successfully 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, 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 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 successfully

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,

	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 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 successfully

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, 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 update the settings.
SourceUserID	User ID that was used to update the settings.
DeviceFacility	Type of settings.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	`tenant ID`
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	`tenant name`

Tenant was successfully 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, 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 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 successfully 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 successfully 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 successfully 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`

Updated data retention policy after changing drives

Event field name	Field value
DeviceAction	`storage policy modified`
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 change the tenant data.
SourceUserID	User ID that was used to change the tenant data.

The dictionary was successfully updated on the service or operation was unsuccessful

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, 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.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	`tenant ID`
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	`tenant name`
Message	If EventOutcome = `failed`, an error message can be found here.

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 login that was used to change the tenant data.
SourceUserID	User ID that was used to change the tenant data.
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`

Response via KICS for Networks

Event field name	Field value
DeviceAction	`KICS 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	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`
DeviceAssetID	Asset ID.
SourceHostName	Asset FQDN.
Name	Asset name.
DeviceCustomString1	List of IP addresses for the asset.
DeviceCustomString1Label	`addresses`

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`

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 3.0.2. It provides the scenarios covered by rules, the conditions of their use, and the necessary sources of events.

> The correlation rules described in this document are contained in the SOC_package file in the KUMA distribution kit; the password for the file is SOC_package1. Only one version of the SOC rule set can be used at a time: either Russian or English.

> You can import correlation rules into KUMA. See the "Importing resources" section of the online help: https://support.kaspersky.com/KUMA/3.0.2/en-US/242787.htm.

> You can add imported correlation rules to correlators that your organization uses. See the online help section "Step 3. Correlation": https://support.kaspersky.com/KUMA/3.0.2/en-US/221168.htm.

Download the description of correlation rules contained in the SOC_package.xlsx file.

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".

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.

To send test events, you need:

- The 'kuma' file running with certain parameters.

  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 KUMA **Task manager** section and click the **Export events** 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 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",
   "name": "-",
   "connection": {
   "name": "-",
   "kind": "tcp",
   "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.

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",
- "1384216367111222333".

Mapping fields of predefined normalizers

The file available via the download link contains a description of the field mapping of preset normalizers.

Download Description of field mapping of preset normalizers.ZIP

Deprecated resources

List of deprecated resources

Name	Resource type	Description
[Deprecated] [OOTB] Microsoft SQL Server xml	Normalizer	Designed for processing events of MS SQL Server versions 2008, 2012, 2014, 2016. This normalizer will be removed from the OOTB set in the next release. If you are using this normalizer, you must migrate to the [OOTB] Microsoft Products normalizer.
[Deprecated] [OOTB] Windows Basic	Normalizer	This normalizer will be removed from the OOTB set in the next release. If you are using this normalizer, you must migrate to the [OOTB] Microsoft Products normalizer.
[Deprecated] [OOTB] Windows Extended v.0.3	Normalizer	This normalizer will be removed from the OOTB set in the next release. If you are using this normalizer, you must migrate to the [OOTB] Microsoft Products normalizer.
[Deprecated] [OOTB] Cisco ASA Extended v 0.1	Normalizer	This normalizer will be removed from the OOTB set in the next release. If you are using this normalizer, you must migrate to the [OOTB] Cisco ASA and IOS syslog normalizer.
[Deprecated] [OOTB] Cisco Basic	Normalizer	This normalizer will be removed from the OOTB set in the next release. If you are using this normalizer, you must migrate to the [OOTB] Cisco ASA and IOS syslog normalizer.

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 is a registered trademark or a trademark of the Apache Software Foundation.

Ubuntu, LTS are registered trademarks of Canonical Ltd.

Cisco, Snort are registered trademarks or trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

Citrix is a trademark of Citrix Systems, Inc. and/or one or more of its subsidiaries, and may be registered in the United States Patent and Trademark Office and in other countries.

The Grafana word mark and the Grafana logo are 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.

Firebird is a registered trademark of the Firebird Foundation.

Fortinet, FortiGate are trademarks or registered trademarks of Fortinet, Inc. in the United States and/or other countries.

The FreeBSD mark is a registered trademark of The FreeBSD Foundation.

Google and Chrome are trademarks of Google LLC.

HUAWEI is a trademark of Huawei Technologies Co., Ltd.

IBM, Guardium, InfoSphere are trademarks of International Business Machines Corporation registered in many jurisdictions around the world.

Intel, Core are trademarks of Intel Corporation in the United States and/or other countries.

Juniper Networks and JUNOS are trademarks or registered trademarks of Juniper Networks, Inc. in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States and other countries.

OpenAPI is a trademark of The Linux Foundation.

Microsoft, Active Directory, Excel, Halo, Hyper-V, PowerShell, SQL Server, Windows, and Windows Server are trademarks of the Microsoft group of companies.

CVE is a registered trademark of The MITRE Corporation.

Mozilla and Firefox are trademarks of the Mozilla Foundation in the United States and other countries.

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.

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.

Symantec is a trademark or registered trademark of Symantec Corporation or its affiliates in the United States 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.

Trend Micro is a trademark or registered trademark of Trend Micro Incorporated.

VMware and VMware ESXi are trademarks of VMware, Inc. or registered trademarks of VMware, Inc. in the United States or other jurisdictions.

ClickHouse is a trademark of YANDEX LLC.

Zabbix is a registered trademark of Zabbix SIA.

ViPNet is a registered trademark of Infotecs.

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 program has been installed and that have been clustered together for centralized management using the program'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 program 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 programs running on the same host to receive the data independently of each other. Each program processes the data sent to a specific port (sometimes it is said that the program 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 a program 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.