kaspersky

Kaspersky Threat Attribution Engine

Contents

About Kaspersky Threat Attribution Engine

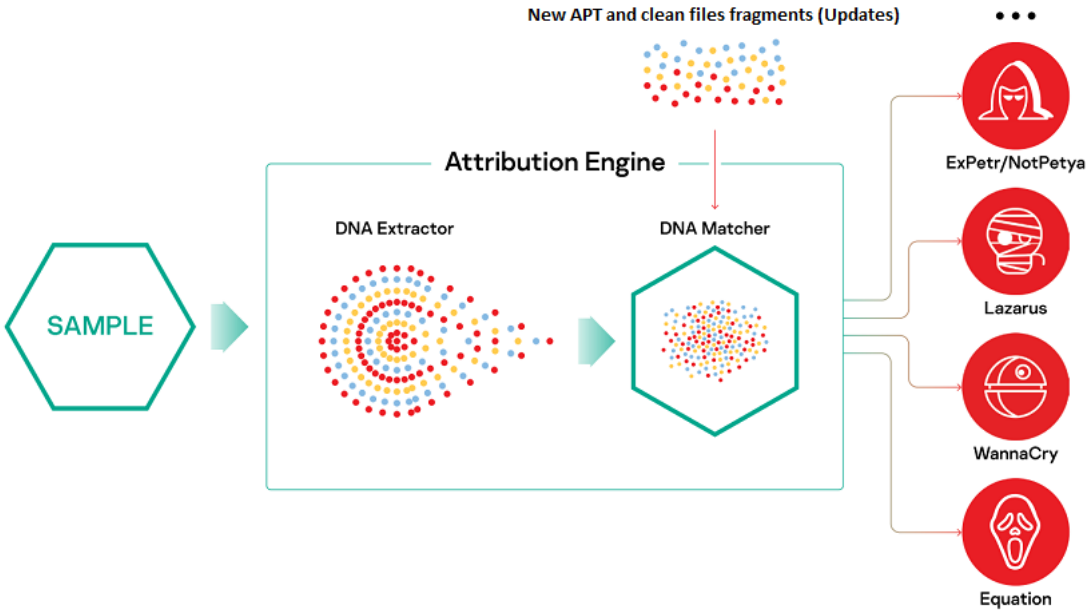
Kaspersky Threat Attribution Engine analyzes a file, or a *sample*, and attributes it to known advanced persistent threat (APT) actors, their tools, related malware, and campaigns. These attribution aspects are combined as *attribution entities* in Kaspersky Threat Attribution Engine.

For example, Kaspersky Threat Attribution Engine attributes a malware sample to a known APT actor only if this actor uses this particular malware. If the analyzed sample file belongs to a popular malware application, it will be attributed to the malware and not to a specific actor.

How Kaspersky Threat Attribution Engine works

To link malware to attribution entities, Kaspersky Threat Attribution Engine uses a unique proprietary method of searching for similarities between files. For this method, Kaspersky Threat Attribution Engine does the following:

1. Analyzes the *genetics* of a sample by extracting the following pieces from its code:
   - *Genotypes*—Distinctive pieces of binary code.
   - *Strings*—Distinctive strings of characters.
2. Automatically searches for the genotypes and strings from the analyzed file that are similar to genotypes and strings of APT samples that were previously analyzed or are already linked to attribution entities.
3. Based on similar genotypes and strings found in APT samples, provides a report on origin of the analyzed sample, related attribution entities, and similarity between the sample and known APT samples.



Kaspersky Threat Attribution Engine workflow

All results obtained from Kaspersky Threat Attribution Engine and created when analyzing these results are of an evaluative nature and cannot be considered or used as evidence. Threats and attribution entities that are classified as APT by Kaspersky Threat Attribution Engine might not necessarily be classified as APT by the user. Kaspersky Threat Attribution Engine cannot be used to detect malicious applications and cyber threats. It is up to the user to make a final decision about the status of any sample or attribution entity.

Key benefits

In addition to the attribution functionality, Kaspersky Threat Attribution Engine provides the following benefits:

- By analyzing genotypes and strings that are extracted from the analyzed sample, Kaspersky Threat Attribution Engine calculates the reputation score of the sample and reveals its genetics and code attribution. This provides insights into the origin of the sample and attributes it to possible authors.
- Security teams can add private attribution entities and related samples to Kaspersky Threat Attribution Engine database. The security teams can educate the application to attribute submitted samples to these private attribution entities and samples.
- With Kaspersky Threat Attribution Engine, the attribution process takes only seconds compared to the years required in the past.

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

Distribution kit

This section describes the Kaspersky Threat Attribution Engine distribution kit.

About the distribution kit

Kaspersky Threat Attribution Engine distribution kit includes the following files and items:

- RPM packages:
  - RPM package for Kaspersky Threat Attribution Engine
  - SafeNet drivers to work with a USB token
- A file with a license key for Kaspersky Threat Attribution Engine
- A set of files containing database data
- A USB token
- Installation scripts
- Set of configuration files

Distribution kit files

The table below lists the files and directories included in the distribution kit.

Distribution kit files

Item	Description
bin/ktae-2.3.0-1.x86_64.rpm	RPM package with Kaspersky Threat Attribution Engine

bin/SafenetAuthenticationClient-9.1.7-0.x86_64.rpm	RPM package with SafeNet driver for the USB token
conf/application.conf	Configuration file with default service parameters
conf/my.cnf	MySQL configuration file with recommended parameters
conf/openssl.cnf	Configuration file for SSL certificate generation
conf/settings.scanner.xml	Configuration file for downloading anti-virus database updates
conf/settings.ktae.xml	Configuration file for Kaspersky Threat Attribution Engine database updates downloading
deps/expect_deps.tgz	Archive that contains utilities for automatic Kaspersky Threat Attribution Engine installation
bin/offline-repo.tar.gz	Archive that contains dependencies for automatic Kaspersky Threat Attribution Engine installation
license/appinfo.kli	Service file with licensing information
license/license.key	Kaspersky Threat Attribution Engine license key file. Make sure to place the license.key file to the license directory to install Kaspersky Threat Attribution Engine.
scripts/auto/*	Scripts for automatic Kaspersky Threat Attribution Engine installation
scripts/sql/*	SQL scripts used during the installation
scripts/check_installation.sh	Shell script used during the installation
scripts/check_installed_packages.sh	Shell script used during the installation
scripts/drop_db.sh	Shell script that deletes the database for reinitialization
scripts/generate_ssl_certificate.sh	Shell script used during the installation
scripts/install_ktae.sh	Shell script used during the installation
scripts/install_safenet_driver.sh	Shell script used during the installation
txt/ktae_data_md5s.txt	File with MD5 hashes of files containing database data
upgrade/scripts/upgrade_config.sh	Shell script used during the upgrade
upgrade/upgrade_ktae.sh	Shell script used to upgrade Kaspersky Threat Attribution Engine from previous versions
upgrade/upgrade_ktae_updater.sh	Shell script used to upgrade the Updater service from previous versions
install_ktae_updater.sh	Shell script that installs the Updater service during the installation
install_and_run_ktae_auto.sh	Shell script that automatically installs dependencies and Kaspersky Threat Attribution Engine and runs it
00_install_dependencies.sh	Shell script that installs dependencies
00_install_dependencies_offline.sh	Shell script that installs prepared dependencies from offline repo without access to the internet
01_install_ktae.sh	Shell script that installs Kaspersky Threat Attribution Engine
02_run_ktae.sh	Shell script that runs Kaspersky Threat Attribution Engine
03_discard_db_tables.sh	Shell script that prepares database tables for import
04_copy_db_tables.sh	Shell script that copies database files to MySQL directory
05_import_db_tables.sh	Shell script that imports database tables from the copied database files

Kaspersky Threat Attribution Engine directory files

The table below lists the files extracted to the `/opt/ktae` directory during the installation.

Kaspersky Threat Attribution Engine directory files

Item	Description
bin/service	Kaspersky Threat Attribution Engine executable file

bin/service_wrap	Executable file for starting Kaspersky Threat Attribution Engine service
bin/admin_cli	Administration utility
cert/server.p12	Server certificate
conf/application.conf	Configuration file
conf/application.ini	Configuration file
conf/logback.xml	Configuration file
docs/legal_notices.txt	Legal notices
docs/license_KTAE_2_3_0_en.rtf	End User License Agreement (EULA)
docs/Release_Notes_KTAE_2_3_0_en.pdf	Release notes
extracted (empty directory)	Directory for extracted genotypes and strings
lib/*	Auxiliary libraries
log/logFile.log	File with Kaspersky Threat Attribution Engine logs
share (empty directory)	Directory for storing unpacked files
tools/linux/binutils/strings	Utility for extracting strings
tools/linux/genotype	Utility for extracting genotypes
tools/linux/scanner/*	Utility for unpacking the compound objects and providing anti-virus decisions for the analyzed files and its files.
uploaded (empty directory)	Directory for downloaded files
integrity_check.xml	File signed with a digital signature that contains a list of unchangeable Kaspersky Threat Attribution Engine files

Updater directory files

The table below lists the files extracted to the `/opt/ktae-updater` directory during the installation.

Updater directory files

Item	Description
bases/*	Directory for downloaded database files
bin/appinfo.kli	Service file with licensing information
bin/cacert.pem	Root certificates of the sources for Kaspersky Threat Attribution Engine update components
bin/keepup2date8	Updater service binary
bin/keepup2date8.sh	Script that runs the Updater service
bin/license.key	Kaspersky Threat Attribution Engine license key file
bin/settings.xml	Configuration file
bin/updatersdk8.log	Log file for the Updater service
bin/updsdk.xml	Configuration file
docs/legal_notices.txt	Legal notices for the application
lib/*	Auxiliary libraries
download_all.sh	Shell script that downloads all Kaspersky Threat Attribution Engine updates
download.sh	Shell script that downloads updates
download_scanner_updates.sh	Shell script that downloads anti-virus database updates
integrity_check.xml	File signed with a digital signature that contains a list of unchangeable Updater service files
settings.scanner.xml	Configuration file for anti-virus database updates downloading
settings.ktae.xml	Configuration file for Kaspersky Threat Attribution Engine database updates downloading

version	File that contains information about Updater service version

Hardware and software requirements

Kaspersky Threat Attribution Engine has the following hardware and software requirements:

Security requirements

- USB token with encryption keys
  This token must be connected to the server where the Kaspersky Threat Attribution Engine will run.
- Server certificate
  By default, a self-signed certificate is used.

General requirements

- Internet connection for working with Kaspersky Threat Attribution Engine
- Open 443 and 80 ports

Minimum hardware requirements

- Intel® Xeon® 2.2 GHz (10C/20T) or a compatible equivalent
- 128 GB of RAM
- 2 TB of free SSD disk space to store the operating system files and the application database
- 1 TB of free HDD disk space to store the application updates

The SSD disk must be used as a system disk. The HDD disk must be used to store the application updates only. We recommend you using Read-Intensive, RAID 5 + spare for both disks.

Supported operating systems

Kaspersky Threat Attribution Engine supports Rocky Linux 9.4 operating system.

Software requirements

- Bash command processor
- OpenSSL
- MySQL 8.0.36

- OpenJDK 1.8.0 (java-1.8.0-openjdk)

Supported browsers

- Mozilla™ Firefox™
- Google Chrome™
- Safari®
- Microsoft Edge®
- Chromium™

About user roles

User role defines what Kaspersky Threat Attribution Engine functionality is available to a user in a web interface or API. The following user roles are available:

- *Regular user*
- *User with administrator rights*

Kaspersky Threat Attribution Engine also has an *administrator* role. Administrators have access to the same functionality as users with administrator rights. In addition, only administrators can install and upgrade Kaspersky Threat Attribution Engine and manage it with admin_cli utility. For administrator-only instructions, see Administering Kaspersky Threat Attribution Engine.

The table below describes what Kaspersky Threat Attribution Engine functionality is available in Kaspersky Threat Attribution Engine for regular users and users with administrator rights.

User roles and allowed actions

Action	Regular user	User with administrator rights
Sending files for analysis and attribution	✓	✓
Viewing recent analysis result	✓ (for the *current* user only)	✓ (for *all* users and *all* analyzed samples)
Exploring detailed analysis report	✓ (for the *current* user only)	✓ (for *all* users and *all* analyzed samples)
Viewing analysis history	✓ (for the *current* user only)	✓ (for *all* users)
Managing attribution entities	✓	✓
Searching analysis results	✓	✓
Exporting analysis result data	✓	✓
Viewing Kaspersky Threat Attribution Engine settings	✓	✓
Working with Kaspersky Threat Attribution Engine REST API	✓	✓
Configuring Kaspersky Threat Attribution Engine settings	–	✓
Resolving update conflicts	–	✓

What's new

Kaspersky Threat Attribution Engine offers the following new features and improvements:

Kaspersky Threat Attribution Engine 2.3.0

- **CentOS 7.9.2009 has been replaced with the Rocky Linux 9.4 operating system** due to the end of vendor support for CentOS. The process of migrating an installed product to a new operating system is outlined in the documentation (Help).
- **The application database is now smaller** due to reduced data smoothing, without affecting its composition or quality.
- **Update download speed has been increased** by optimizing snapshots.
- **Improved application stability** by fixing freezes that occurred when analyzing archives containing a large number of files.
- Resolved the issue of unavailable analysis results after license expiration.
- **Documentation** (Help) is now available on the Kaspersky Online Help portal.
- Minor bugfix.
- Documentation enhancement.

Kaspersky Threat Attribution Engine 2.2.0

- **Improved installation and update processes:**
  - All required dependencies can be installed from the Kaspersky Threat Attribution Engine distributive using an automated script, which does not require internet connection.
  - During database update, the log is printed to a terminal so that you can track the update progress.
  - Files signed with a digital signature are added to the distribution package to verify that all files are consistent and issued by Kaspersky.
- **Added functionality of unpacking password-protected archives with custom passwords**. Now, you can specify a custom password when attaching an archived sample to an attribution entity or uploading an archived sample for analysis. In this case, the files will be automatically extracted from the archive using the specified password (in addition to default passwords) before they are attached or analyzed. This functionality is also added to the API as a parameter for the corresponding methods (*/api/samples/{entityName}* and */api/analysis*).
- **Added attachment date parameter for all samples submitted by a user**. The attachment date indicates when the sample was attached to an attribution entity.
- **Added tooltips to web interface elements**. The tooltips can help you understand what you can do on each page.
- **Added links to documentation and End User License Agreement (EULA) to the web interface**. The documentation and EULA can help you understand functionality and license limitations. The documentation

also contains a link to a Kaspersky Threat Attribution Engine lifecycle page, which includes End of Support dates for different versions.

- **Optimized and enhanced documentation** with clarified key concepts and more usage instructions.
- Minor bug fixing.

Kaspersky Threat Attribution Engine 2.1.0

- **Implemented common threat detection**. By using the Anti-Virus engine, you can now attribute common threats in addition to APT-related threats. Detection names for common threats are provided in the web interface and API. Detailed descriptions of detection names are available in the https://threats.kaspersky.com/ portal.
- **Improved exporting capabilities**. You can now export results from the **Analysis**, **History**, analysis result, genotype property, and string property pages. You can also select items for exporting.
- **Implemented functionality for obtaining information about the current license from the REST API** using the new *get license* method.
- **Implemented capability for attributing in parallel files sent for analysis in one request**. This functionality helps to decrease time required to analyze a batch of files.
- **Decreased storage requirements** without any impact on the quality and coverage. See the Hardware and software requirements section in the documentation for more information.
- **Optimized license quota calculation**. When you send compound objects (for example, PDF or Microsoft® Office documents) for analysis with the unpacking option enabled, the license quota is decreased by one unit for each file, and not by the total number of extracted objects. This functionality is not applied to archives, for which every extracted object is considered as a separate quota unit.
- **Enhanced documentation** with descriptions of new functionality.
- Minor bug fixing.

Kaspersky Threat Attribution Engine 2.0.0

- **Implemented analysis history**. The file analysis results are now saved in Kaspersky Threat Attribution Engine for 180 days. You can access history to search previously analyzed files (submitted from web interface or API), export reports, estimate load metrics and available license analysis limit, and identify active users. Regular users can view only analysis history that they submitted, users with administrator rights can view history of all users.
- **Implemented highlighting similarities between submitted files**. This feature helps you understand if an analyzed file is similar to any previously analyzed files and how they are related. The highlighted similarity is available in the **Previously matched samples** column on the **Analysis** and **History** pages. You can click the column value to switch to a detailed report showing common genotypes and strings.
- **Implemented new exporting capabilities**:
  - You can export genotypes and strings of the analyzed files to YARA rules for further automated search/scanning for similar files (threat hunting) or integration with third-party solutions. Simple and cluster rules are supported.
  - You can export file analysis results to STIX™ 2.1 format for further automated analysis of security logs (in SIEM, for example) or integration with third-party solutions and security controls.

- **Added new capabilities to the REST API**, such as asynchronous analysis mode, batch analysis mode, heartbeat, timestamps, and version information.
- **Enhanced documentation** with descriptions of new functionality.
- Minor bug fixing.

Kaspersky Threat Attribution Engine 1.0.1

- **Enabled Amazon Web Services™ (AWS™) cloud support**. You can now deploy Kaspersky Threat Attribution Engine on AWS. AWS installation gives you benefits of cloud infrastructure, including quick setup and cost saving, as you do not need to invest into hardware and infrastructure upfront. With the help of an Amazon™ Machine Image (AMI) provided by Kaspersky, you can get a working installation of Kaspersky Threat Attribution Engine in less than an hour.
- Minor bug fixing.

Kaspersky Threat Attribution Engine 1.0.0

- **Implemented file analysis for threat attribution**. You can upload a file (sample) using the web interface or API to automatically analyze the file for similarity with previously investigated APT samples and related attribution entities. Kaspersky Threat Attribution Engine extracts genotypes and strings from the decompiled file, compares them with the Kaspersky database of APT malware samples, and calculates file reputation score. Based on sample genotypes and code attribution, Kaspersky Threat Attribution Engine provides you with information about similarity between the analyzed sample and known attribution entities (actors, campaigns, or threats), as well as their descriptions.
- **Implemented education mode**. You can create a new attribution entity and attach your own samples to this entity. Genotypes of the samples are added to a private database. Kaspersky Threat Attribution Engine can attribute new files submitted for analysis to this new entity. Private files can also be attributed with any existing attribution entities. You can delete the private samples as well.
- **Implemented attribution entity management**. You can create or delete attribution entities. Every attribution entity has the following fields:
  - **Name**—Attribution entity name.
  - **Parent**—Name of the parent attribution entity.
  - **Genotypes threshold**—Minimum number of genotypes that must match to attribute a sample to the attribution entity.
  - **Strings threshold**—Minimum number of strings that must match to attribute a sample to the attribution entity.
  - **Description**—Attribution entity description.
  - **Aliases**—List of known aliases for the attribution entity.
  - **Report links**—Links to reports associated with the attribution entity.
  - **Comments**—Miscellaneous information about the attribution entity.

  You can modify the fields for custom attribution entities. For Kaspersky attribution entities, which are preset, you can only modify the **Comments** field.

For custom attribution entities, you can also adjust similarity level that determines how many genotypes/strings must match to consider files similar.

- **Implemented exporting capability**. You can export file analysis results to TXT and JSON formats for further automated analysis or integration with third-party solutions.

- **Integrated Kaspersky DeepUnpack**. Kaspersky DeepUnpack technology provides possibility of handling compressed objects. Kaspersky DeepUnpack supports more than 4000 packer and archive formats. It allows to retrieve genotypes and strings of an original object even if it was packed or archived using different formats (to be obfuscated). Kaspersky DeepUnpack is used when a packed file is submitted for analysis or attached to an attribution entity.

- **Enabled support for any binary data**. Kaspersky Threat Attribution Engine can analyze all types of executable files, as well as other file types such as PDF, Microsoft Word documents, Java scripts, dump files, or certificate files.

- **Enabled on-premise deployment**. Kaspersky Threat Attribution Engine can be deployed in a secure air-gapped environment, which prevents any third party from accessing processed information and submitted objects.

- **Enabled database updates**. Regular updates are supported to keep Kaspersky Threat Attribution Engine up to date with data about new APT threats (genotypes, strings, Kaspersky attribution entities).

- **Enabled allowlist support**. The biggest in the industry database of legitimate files is supported and updated to exclude false positives and provide high-quality similarity level. The machine learning approach is used to consider genotypes and strings of legitimate files when making the attribution.

- **Enabled RESTful API**. Kaspersky Threat Attribution Engine has an API to connect to other tools and frameworks and implement attribution to existing infrastructure and automated processes. The following scenarios are supported:

  - Analyzing a file for threat attribution

  - Getting file analysis and attribution results

  - Creating an attribution entity

  - Attributing a file to an attribution entity (education mode)

- **Enabled role-based access control (RBAC)**. Kaspersky Threat Attribution Engine supports two roles:

  - *Regular users* can submit files for analysis and attribution, create and manage their own attribution entities, associate private samples with any attribution entity.

  - *Users with administrator rights* can also remove attribution entities created by other users, access a log of files that were associated with attribution entities, exclude legitimate files from attribution (in cases when a user tries to associate a legitimate file with attribution entities).

- **Added documentation** in HTML format.

Installing Kaspersky Threat Attribution Engine

This section explains how to install and configure Kaspersky Threat Attribution Engine.

Common deployment scenarios

This section provides a description of standard deployment schemes of Kaspersky Threat Attribution Engine in an enterprise network.

> Fresh Kaspersky Threat Attribution Engine installation on Rocky 9.4 and migration of the existing application instance from CentOS 7 to Rocky 9.4 are described in the corresponding sections.

Kaspersky Threat Attribution Engine includes the following services:

- Kaspersky Threat Attribution Engine service—Main service that provides all functionality of Kaspersky Threat Attribution Engine.
- Updater service—Additional service that downloads updates for Kaspersky Threat Attribution Engine.

Deployment scenarios

You can install Kaspersky Threat Attribution Engine using one of the following configurations:

- **Simple configuration**—Install Kaspersky Threat Attribution Engine and Updater services on a single server that has access to the internet (*open perimeter*).
- **Air-gapped configuration**—Install Kaspersky Threat Attribution Engine and Updater services in different perimeters:
  - Install the Updater service in an *open perimeter* that has access to the internet.
  - Install Kaspersky Threat Attribution Engine service in a *secure perimeter* that is separated from the open perimeter with an air gap.

Installation workflow

Kaspersky Threat Attribution Engine installation includes the following stages:

1. Installing and configuring Rocky Linux 9.4 operating system.
2. Preparing for installation by creating directories to store the update files and creating a special user.
3. Installing the Updater service.
4. Installing the Kaspersky Threat Attribution Engine service.

Installing and configuring Rocky Linux 9.4

This section explains how to install and configure the Rocky Linux 9.4 operating system.

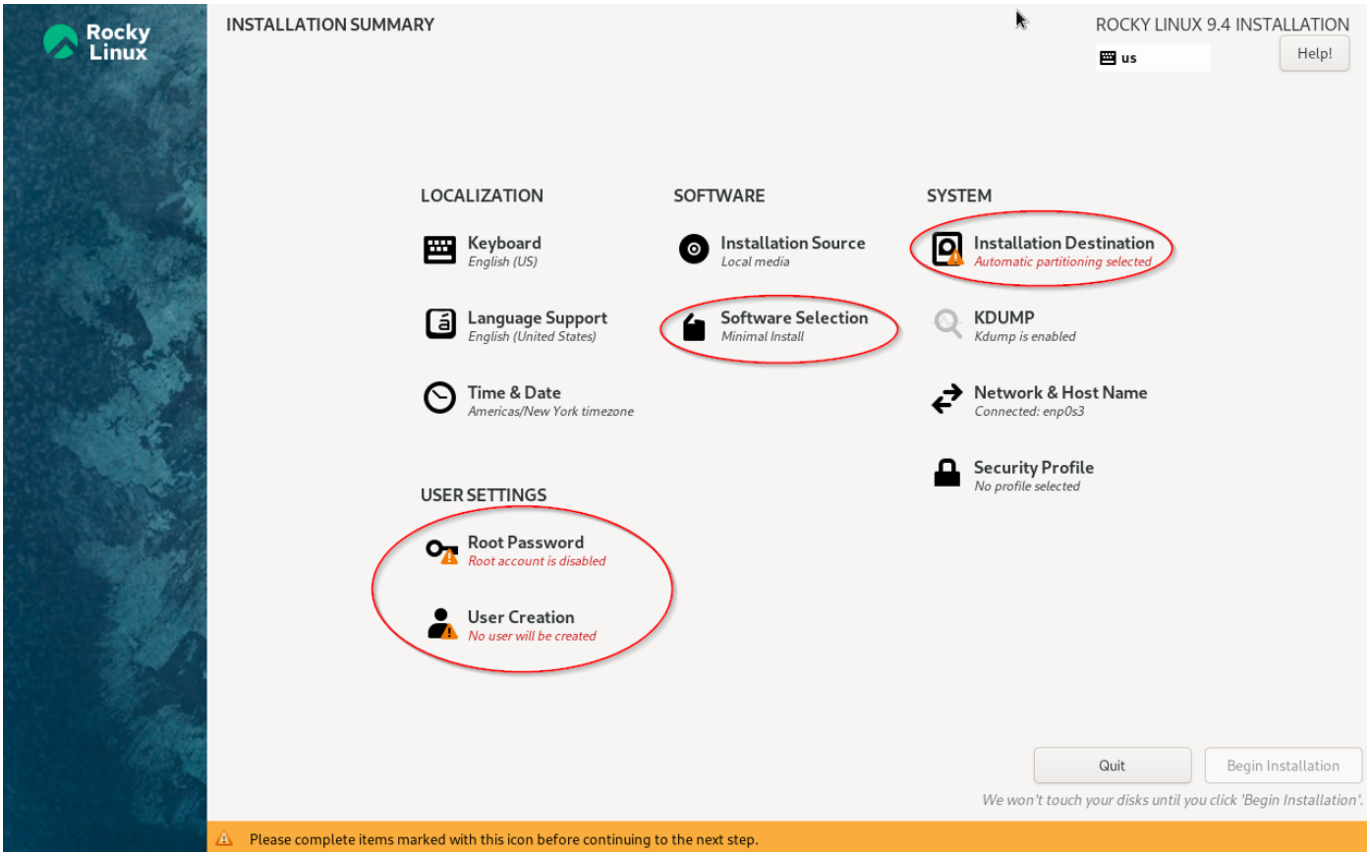
To install Kaspersky Threat Attribution Engine on the Rocky Linux 9.4 operating system:

Download Rocky Linux 9.4 from https://dl.rockylinux.org/vault/rocky/9.4/isos/x86_64/, choose Rocky-9.4-x86_64-minimal.iso.

During the installation process, pay attention to user settings, software selection, and installation destination steps described below.

1. In the **User Settings** section, create a user: fill in the name and password.

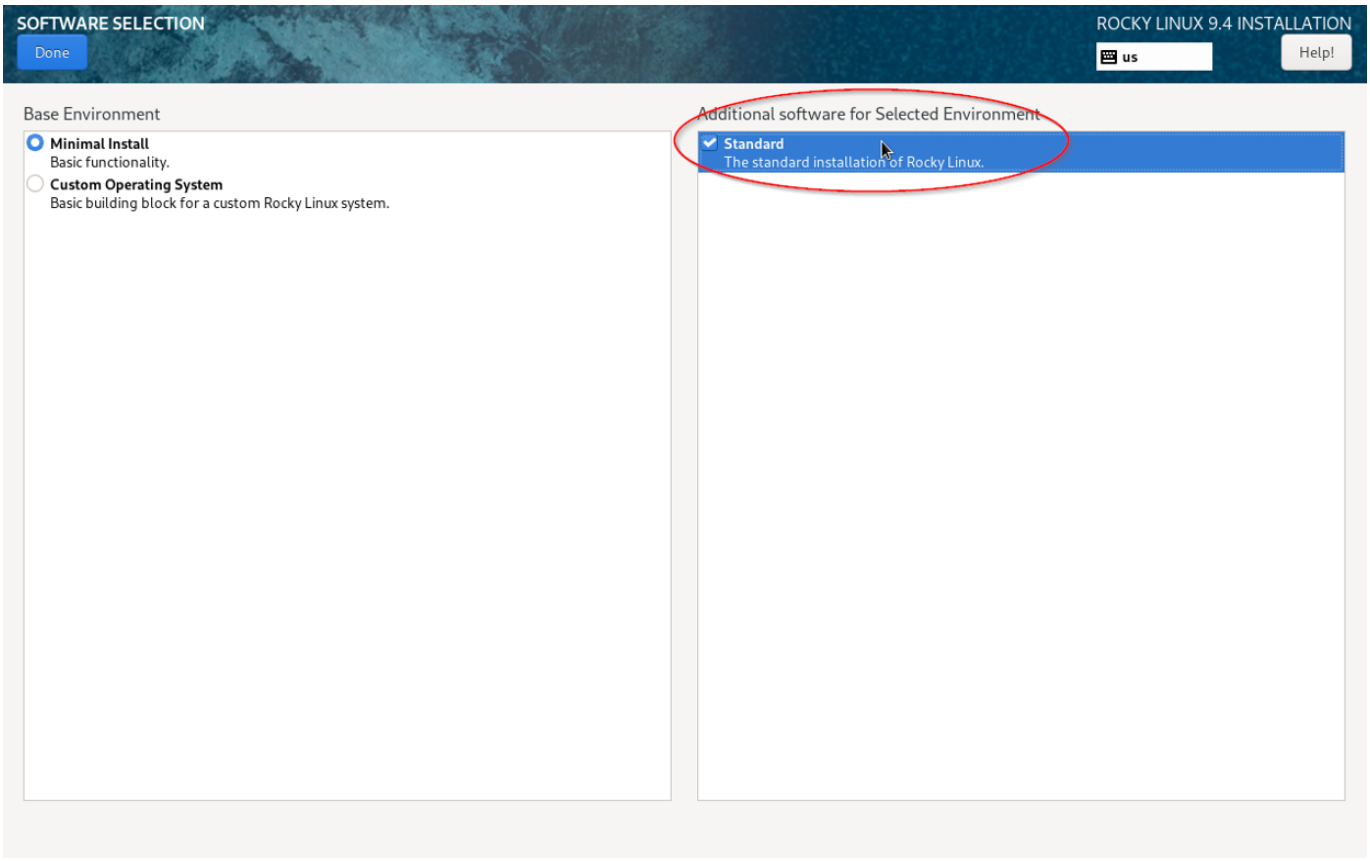
2. Fill in a password for root account.



Rocky Linux 9.4 Installation Summary window

3. In the **Software** section, select additional software for the operating system.

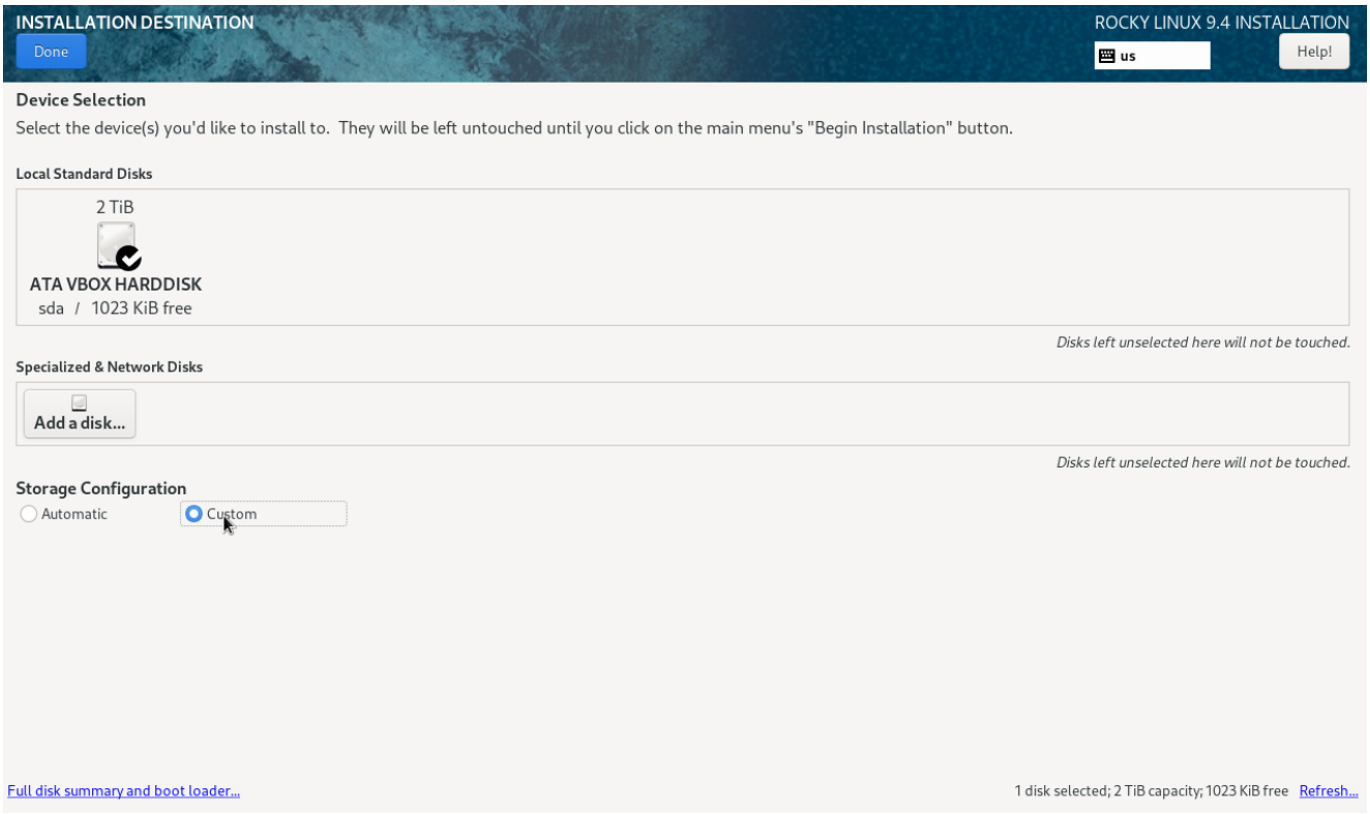
4. **Choose standard software**:



Rocky Linux 9.4 Software Selection window

5. In the **System** section, select the **Installation Destination** option. Check your machine's storage under the **Local Standard Disks** heading. Rocky Linux 9.4 will be installed on the selected disk.

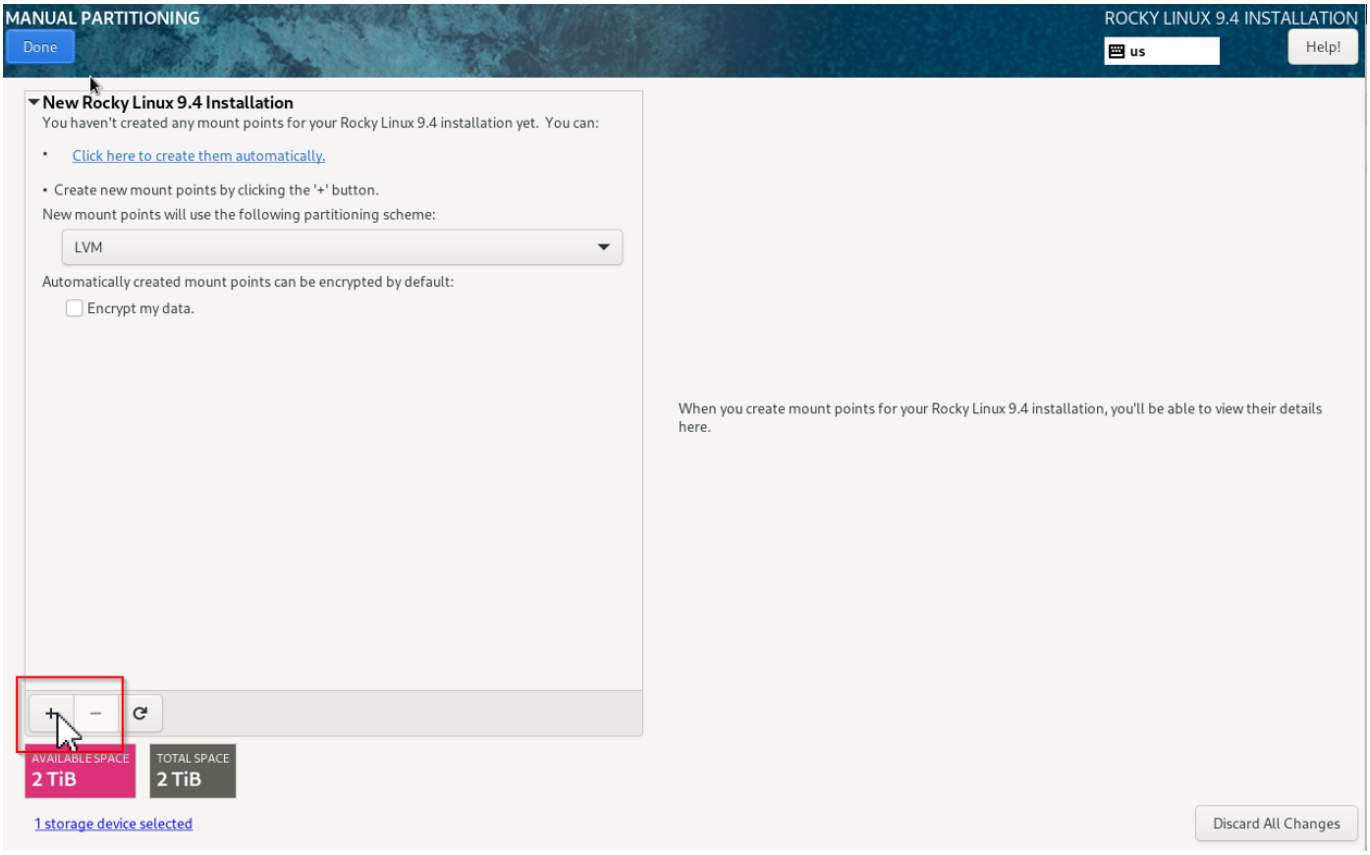
6. **Important!** Under the **Storage Configuration** heading, select the **Custom** option:



Rocky Linux 9.4 Installation Destination window

7. In the **Manual Partitioning** step, create two LVM partitions as follows:

- 2 TB for system (/)
- 1 TB for updates (/update)
- 1 EFI partition for /boot/efi (300 MB)
- 1 XFS partition for ./boot (1 GB)



Rocky Linux 9.4 Manual Partitioning window

8. After completing all configuration steps, reboot the OS and continue installation.

Preparing to install Kaspersky Threat Attribution Engine

This section explains how to prepare perimeters for Kaspersky Threat Attribution Engine installation.

Before installing Kaspersky Threat Attribution Engine, make sure you have at least 1 TB of free HDD disk to store the update files.

See Hardware and software requirements for more information.

Kaspersky Threat Attribution Engine distribution kit includes a package with all required dependencies. The dependencies are installed automatically before Kaspersky Threat Attribution Engine installation. If you need to update the operating system, we recommend that you do this *only after* the installation is completed. Updating the operating system before installing Kaspersky Threat Attribution Engine may cause problems with the installation script.

Preparing to install with a simple configuration

To install Kaspersky Threat Attribution Engine services using a simple configuration, you need to run the preparation steps *only* in the open perimeter.

To prepare the perimeter for installing Kaspersky Threat Attribution Engine:

1. Create users with user names ktae and ktae_updater as follows:

   a. Create users with user names ktae and ktae_updater:

   ```
   sudo adduser -s /bin/bash ktae
   sudo adduser -s /bin/bash ktae_updater
   ```

   b. Add passwords for created users:

   ```
   sudo passwd ktae
   sudo passwd ktae_updater
   ```

   > It is important to create the ktae user before installing Kaspersky Threat Attribution Engine and make sure that only administrator has access to this account.

2. Create a temporary storage directory for Kaspersky Threat Attribution Engine update files and save the path to it to use it during installation.

   In the installation instructions, this directory is named `<TEMP_UPDATES_FILES_DIR>`.

3. Create the main storage directory for Kaspersky Threat Attribution Engine update files and save the path to it to use it during installation.

   In the installation instructions, this directory is named `<OPEN_UPDATES_DIR>`.

4. Before beginning the installation process, copy all files of the distributive (distribution kit) to the directory to which only ktae and root users have read, write, and delete access. All further installation actions must be carried out in this directory only.

For security reasons, there is only a limited set of paths allowed to be used as `<OPEN_UPDATES_DIR>` and `<SECURE_UPDATES_DIR>` (see the next section "*Preparing to install with an air-gapped configuration*").
By default, the following set of paths' prefixes are allowed:

```
/opt/ktae/db/...
/tmp/ktae/...
/mnt/ktae/..
/update/...
/opt/ktae/tools/linux/update/...
```

If you need to use any other directory that does not start with any of these prefixes, you need to add it to the list in /opt/ktae-updater/update_persmissions_for_updates_folder.sh script after you finish installing the Updater service.

Preparing to install with an air-gapped configuration

To install Kaspersky Threat Attribution Engine services using an air-gapped configuration, you need to run the preparation steps in both open and secure perimeters.

To prepare the perimeters for installing Kaspersky Threat Attribution Engine:

1. In *both* open and secure perimeters, create users with user names ktae and ktae_updater as follows:

   a. Create users with user names ktae and ktae_updater:

   ```
   sudo adduser -s /bin/bash ktae
   sudo adduser -s /bin/bash ktae_updater
   ```

   b. Add passwords for created users:

   ```
   sudo passwd ktae
   sudo passwd ktae_updater
   ```

   > It is important to create the ktae user before installing Kaspersky Threat Attribution Engine and make sure that only administrator has access to this account.

2. In the open perimeter, create a temporary storage directory for Kaspersky Threat Attribution Engine update files and save the path to it to use it during installation.

   In the installation instructions, this directory is named `<TEMP_UPDATES_FILES_DIR>`.

3. In the open perimeter, create a storage directory for Kaspersky Threat Attribution Engine update files and save the path to it to use it during installation.

   In the installation instructions, this directory is named `<OPEN_UPDATES_DIR>`.

4. In the secure perimeter, create the main storage directory for Kaspersky Threat Attribution Engine update files and save the path to it to use it during installation.

   In the installation instructions, this directory is named `<SECURE_UPDATES_DIR>`.

5. Before beginning the installation process, copy all files of the distributive (distribution kit) to the directory to which only ktae and root users have read, write, and delete access. All further installation actions must be carried out in this directory only.

Installing the Updater service

This section explains how to install the Updater service in the open perimeter. These instructions are the same for the simple and air-gapped configurations.

Before installing the Updater service, make sure the perimeter has access to the internet. If the perimeter is protected by a firewall, the following ports must be opened:

- 443—For downloading Kaspersky Threat Attribution Engine updates.
- 80—For downloading the certificate revocation list.

To install the Updater service:

1. Run the install_ktae_updater.sh script:

   ```
   sudo bash install_ktae_updater.sh
   ```

2. When asked, enter a path to the `<TEMP_UPDATES_FILES_DIR>` directory created when preparing for the installation.

3. When asked, type `2` and press **Enter** to select the `Install KTAE updater` option.

4. Read the End User License Agreement (EULA) and do one of the following:

   - Type `Y` and press **Enter** to accept the EULA.

     The installation process starts. Wait until the installation is complete.

   - Type `N` and press **Enter** to decline the EULA.

     The installation process stops.

5. After the installation is complete, change the current working directory to /opt/ktae-updater:

   ```
   cd /opt/ktae-updater
   ```

   This is the directory where the Updater was installed.

6. Run the following command to download the Kaspersky Threat Attribution Engine updates:

   ```
   sudo bash download_all.sh <OPEN_UPDATES_DIR>
   ```

   Replace `<OPEN_UPDATES_DIR>` with a path to the directory created when preparing for the installation. The update files are downloaded to this directory.

   The download process can be time consuming. Wait for the downloading to complete.

   Do not skip step 6. Update files are required for installing Kaspersky Threat Attribution Engine.

You successfully installed the Updater service and downloaded the required update files. Proceed to install the Kaspersky Threat Attribution Engine service.

Installing and configuring the Kaspersky Threat Attribution Engine service

This section explains how to install and configure the Kaspersky Threat Attribution Engine service and its dependencies.

Installing the Kaspersky Threat Attribution Engine service

Before you start Kaspersky Threat Attribution Engine installation, make sure that the following prerequisites are met:
1. You are logged into the system as a root user.
2. USB token with encryption keys is connected to your server (see the "Hardware and software requirements" section for more information).

To install the Kaspersky Threat Attribution Engine service and its dependencies, you can use an automated script install_and_run_ktae_auto.sh. Depending on the configuration you used to install Kaspersky Threat Attribution Engine, you should run the script in different perimeters:

- For the simple configuration, run the script in the open perimeter where you installed the Updater service.
- For the air-gapped configuration, run the script in the separate secure perimeter.

Installing the Kaspersky Threat Attribution Engine service

To install the Kaspersky Threat Attribution Engine service and its dependencies:

1. Open a terminal and navigate to the distribution kit directory.
2. Run the following script to start the installation process:

   ```
   sudo bash install_and_run_ktae_auto.sh
   ```

3. Read the End User License Agreement (EULA) printed to the terminal and do one of the following:
   - Type **Y** and press **ENTER** to accept the EULA and proceed to installation.
   - Type **N** and press **ENTER** to decline the EULA and stop installation.
4. When prompted, enter a new password for a root user of MySQL, which was created by default.

   The root user is used for the Kaspersky Threat Attribution Engine database schema and the SQL script that creates users.

   The password must contain the following characters to meet the minimum requirements for a strong password:
   - 9 characters
   - 2 uppercase letters
   - 2 lowercase letters
   - 2 numbers
   - 2 of the following allowed special characters: ‘ ~ ! @ # $ % ^ & * ( ) _ - + = { } [ ] / < > , . ; ? ' : | and space
5. When prompted, create a password for a ktae_user user of MySQL, which will be automatically created later.

   The ktae_user user is used to access MySQL.

   The passwords must meet the same requirements as the root user password (described in the step above).

The script automatically installs the required dependencies and the Kaspersky Threat Attribution Engine service and generates a self-signed SSL certificate. The certificate will be used for accessing the application by default. Kaspersky Threat Attribution Engine service is installed to the /opt/ktae directory. For full installation log, see the /tmp/ktae_auto_installation.log file.

After installation is completed, the script starts Kaspersky Threat Attribution Engine and MySQL server and prints their working status to the terminal. Make sure the MySQL server and Kaspersky Threat Attribution Engine service have the `active (running)` status.

Verifying the installation

To verify the installation:

1. From the terminal, run the following command to make sure that the USB token is available:

   ```
   lsusb
   ```

   The token name should be present in the list as `0529:0620 Aladdin Knowledge Systems Token JC`.
2. Make sure that the following directories have ktae:ktae as the owner:

- /opt/ktae
- /opt/ktae/uploaded
- /opt/ktae/extracted

3. Make sure that the genotype and strings utilities have an execute permission.

   The utilities are located in the following directories:

   - genotype utility in opt/ktae/tools/linux
   - strings utility in /opt/ktae/tools/linux/binutils

4. Make sure that the only ktae user has read permission to access the default SSL certificate that is automatically generated at /opt/ktae/cert/server.p12.

   This certificate is used by default. However, we recommend that you use a custom certificate with Kaspersky Threat Attribution Engine. In this case, make sure the ktae user has read access to the custom certificate.

5. If you have an anti-virus installed, make sure that the following directories are added to the anti-virus exceptions:

   - /opt/ktae/uploaded
   - /opt/ktae/extracted
   - /opt/ktae/share

You successfully installed the Kaspersky Threat Attribution Engine service. Proceed to configure the service.

Configuring the Kaspersky Threat Attribution Engine service

Depending on the configuration you used to install Kaspersky Threat Attribution Engine, you should run the configuration steps in different perimeters:

- For the simple configuration, run the steps in the open perimeter.
- For the air-gapped configuration, run the steps in the separate secure perimeter.

Configuring the Kaspersky Threat Attribution Engine service includes the following stages:

1. Importing a Kaspersky Threat Attribution Engine database to MySQL.
2. Configuring Kaspersky Threat Attribution Engine updates and updating the anti-virus databases.
3. Specifying an origin and domain names for Kaspersky Threat Attribution Engine portal.

Follow the instructions below to complete these stages.

Configuring MySQL

The Kaspersky Threat Attribution Engine service installation script applies the recommended configuration of MySQL. If you need to change the default configuration provided by the installation script, perform steps described below.

To configure MySQL:

1. Stop the MySQL service:
   ```
   systemctl stop mysqld
   ```

2. Open the /etc/my.cnf file and make sure it looks like following:
   ```
   [mysqld]
   local_infile=1
   tmp_table_size=4G
   max_heap_table_size=4G
   [mysqld]
   bind-address = 0.0.0.0
   port = 3306
   disable_log_bin
   key_buffer_size=32M
   default-storage-engine=innodb
   innodb_buffer_pool_size=64GB
   innodb_buffer_pool_instances=64
   innodb_log_buffer_size=8M
   innodb_write_io_threads=64
   innodb_read_io_threads=64
   innodb_parallel_read_threads=64
   innodb_io_capacity=100000
   innodb_change_buffer_max_size=50
   innodb_compression_level=9
   innodb_lock_wait_timeout=100
   innodb_log_file_size=512M
   innodb_file_per_table
   innodb_flush_log_at_trx_commit=0
   datadir=/var/lib/mysql
   socket=/var/lib/mysql/mysql.sock
   log-error=/var/log/mysqld.log
   pid-file=/var/run/mysqld/mysqld.pid
   ```

3. After you have made the necessary changes, save and close the my.cnf file, and then start the MySQL service:
   ```
   systemctl start mysqld
   ```

Importing a Kaspersky Threat Attribution Engine database to MySQL

To import the database to MySQL:

1. Run the 03_discard_db_tables.sh script to prepare the Kaspersky Threat Attribution Engine database for importing:
   ```
   sudo bash 03_discard_db_tables.sh
   ```

2. When asked, enter root user password for MySQL.

3. Compare the MD5 hashes of the database information files to MD5 hashes in the txt/ktae_data_md5s.txt file and make sure the hashes are identical.

4. Run the 04_copy_db_tables.sh script to copy the database tables:
   ```
   sudo bash 04_copy_db_tables.sh
   ```

5. When asked, enter the path to the directory that contains database files in the IBD format.

6. Run the 05_import_db_tables.sh script to import the database:

```
sudo bash 05_import_db_tables.sh
```

7. When asked, enter root user password for MySQL.

Updating the anti-virus databases

To configure Kaspersky Threat Attribution Engine updates and update the anti-virus databases:

1. If you installed the Kaspersky Threat Attribution Engine in the secure perimeter (air-gapped configuration), copy the update files from the `<OPEN_UPDATES_DIR>` directory in the open perimeter to the `<SECURE_UPDATES_DIR>` directory in the secure perimeter.

   The update files were downloaded with the Updater service using the download_all.sh script.

2. Make sure that the created update directory exists and has `ktae_updater:ktae` as its owner.

3. Open the /opt/ktae/conf/application.conf file and specify a path to the update directory in the `path` parameter of the `database` > `update` section. Kaspersky Threat Attribution Engine will look for update files in this directory.

   Depending on the configuration, specify one of the following directories:

   - For the simple configuration, specify a path to the directory in the open perimeter `<OPEN_UPDATES_DIR>`:

     ```
     database {
     ...
       updates {
         path = "<OPEN_UPDATES_DIR>" // <-- specify your update directory here
       }
     }
     ```

   - For the air-gapped configuration, specify a path to the directory in the secure perimeter `<SECURE_UPDATES_DIR>` to which you copied the update files:

     ```
     database {
     ...
       updates {
         path = "<SECURE_UPDATES_DIR>" // <-- specify your update directory here
       }
     }
     ```

4. Launch the admin_cli utility:

   ```
   sudo bash /opt/ktae/bin/admin_cli
   ```

5. Run the following command to stop the Kaspersky Threat Attribution Engine service:

   ```
   terminate-service
   ```

6. Run the following command to update the anti-virus databases:

   ```
   activate-scanner-bases
   ```

   If you used the simple configuration, you can configure anti-virus databases for automatic updates and skip this step.

7. Start the Kaspersky Threat Attribution Engine service to verify the updates:

```
run-service
```

Specifying an origin and domain names

To change default origin and domain names,

Open the /opt/ktae/conf/application.conf file and specify a new origin and domain names in the `auth-settings` section. These origin and domain names will be used to access the Kaspersky Threat Attribution Engine web portal and API.

> If you are going to run Kaspersky Threat Attribution Engine on the same machine where you installed in (that is, on a localhost), you can keep the default domain and origin names. However, we do not recommend it.

You can use an IP address or a domain. For example:

- If you installed Kaspersky Threat Attribution Engine on a machine with 1.1.1.1 IP address, specify it in the file as follows:

```
service-settings {
  host = "0.0.0.0"
  port = 8080

  auth-settings {
    origin-name = "https://1.1.1.1:8080" // use KTAE IP address
    domain-name = "1.1.1.1"              // KTAE IP address
  }
  ...
}
```

- If you installed Kaspersky Threat Attribution Engine on a machine with ktae.mydomain.org domain, specify it in the file as follows:

```
service-settings {
  host = "0.0.0.0"
  port = 8080

  auth-settings {
    origin-name = "https://ktae.mydomain.org:8080" // use KTAE domain
    domain-name = "ktae.mydomain.org"              // KTAE domain
  }
  ...
}
```

In other sections of the document, the Kaspersky Threat Attribution Engine IP address and domain will be referred as `<KTAE_IP_ADDRESS>` and `<KTAE_DOMAIN>`.

You successfully installed and configured Kaspersky Threat Attribution Engine and its dependencies.

Migration of existing instance from CentOS 7 to Rocky Linux 9.4

This section explains how to migrate an existing Kaspersky Threat Attribution Engine instance from CentOS 7 to Rocky Linux 9.4.

The scenario for migrating Kaspersky Threat Attribution Engine to the Rocky Linux 9.4 operating system consists of the following steps:

1. **Update of Kaspersky Threat Attribution Engine database on the CentOS**

   Before you start migration from CentOS to Rocky Linux 9.4, you must update your current Kaspersky Threat Attribution Engine database.

   > Make sure that the last update date of your Kaspersky Threat Attribution Engine database is not older than February, 2025. Otherwise, all later updates will not be applicable.

   You can check the Kaspersky Threat Attribution Engine version in the web interface (**Settings** → **About**) or using API.

2. **Backup of Kaspersky Threat Attribution Engine files**

   Back up Kaspersky Threat Attribution Engine files on the CentOS.

   Click to expand and view details.

   1. Stop ktae service:
      ```
      # systemctl stop ktae
      ```
   2. Copy your configuration file /opt/ktae/conf/application.conf to a separate disk on another server.
   3. Make sure that mysql service is up and running:
      ```
      systemctl start mysqld
      ```
   4. From a directory with Kaspersky Threat Attribution Engine distributive, run the automated script to backup your MySQL database:
      ```
      bash ./scripts/backup_mysql_db.sh <output_directory>
      ```
      You will be prompted to enter you MySQL root password.

   This operation may take a long time. When it is finished, you will have a backup of your Kaspersky Threat Attribution Engine database in your <output_directory>.

3. **Installation and configuration of Rocky Linux 9.4**

   Install and configure the Rocky Linux 9.4 operating system.

4. **Preparing for installation in the secure perimeter**

   Prepare to install Kaspersky Threat Attribution Engine in the secure perimeter by installing the necessary dependencies.

   Click to expand and view details.

1. Log in to the system under the user you created in the previous step (*Installation and configuration of Rocky Linux 9.4 operating system*).

2. Change directory to /var/tmp:
   ```
   # cd /var/tmp
   ```

3. Create the ktae directory and switch to it
   ```
   # sudo mkdir ktae && cd ktae
   ```

4. Unpack the archive with Kaspersky Threat Attribution Engine into the /var/tmp/ktae/dist directory.

5. Create ktae and ktae_updater users:
   ```
   # adduser -s /bin/bash ktae
   # adduser -s /bin/bash ktae_updater
   # passwd ktae  (choose password for ktae user)
   # passwd ktae_updater  (choose password for ktae_updater user)
   ```

6. Go to folder:
   ```
   # cd /var/tmp/ktae/dist
   ```

7. Change all scripts to executable:
   ```
   # sudo chmod +x *.sh
   # sudo chmod +x ./scripts/*.sh
   ```

5 Kaspersky Threat Attribution Engine Updater installation

Install the Kaspersky Threat Attribution Engine Updater.

Click to expand and view details.

Before installing the Updater service, make sure the perimeter has access to the internet. If the perimeter is protected by a firewall, the following ports must be opened:

- 443—For downloading Kaspersky Threat Attribution Engine updates.
- 80—For downloading the certificate revocation list.

To install the Updater service:

1. Run the install_ktae_updater.sh script:
   ```
   sudo bash install_ktae_updater.sh
   ```

2. When asked, enter a path to the `<TEMP_UPDATES_FILES_DIR>` directory created when preparing for the installation.

3. When asked, type `2` and press **`Enter`** to select the `Install KTAE updater` option.

4. Read the End User License Agreement (EULA) and do one of the following:

   - Type `Y` and press **`Enter`** to accept the EULA.

     The installation process starts. Wait until the installation is complete.

   - Type `N` and press **`Enter`** to decline the EULA.

     The installation process stops.

6 **Kaspersky Threat Attribution Engine installation**

Install Kaspersky Threat Attribution Engine.

Click to expand and view details.

1. Execute the script install_and_run_ktae_auto.sh.

2. Stop Kaspersky Threat Attribution Engine service:
   ```
   sudo systemctl stop ktae
   ```

During the execution of this script, you will be prompted to enter the root MySQL password — use the one from your CentOS backup.

7 **Restoring MySQL database**

For this step, restore the MySQL database.

Click to expand and view details.

1. Prepare your backup copy of Kaspersky Threat Attribution Engine MySQL database (see the "*Backup of Kaspersky Threat Attribution Engine files*" step).

2. From a directory with Kaspersky Threat Attribution Engine distributive, run the automated script to restore your MySQL database:
   ```
   bash ./scripts/restore_mysql_dump.sh <path_to_dump_archive>
   ```
   You will be prompted to enter you MySQL root password.

This operation may take a long time.

8 **Restoring configuration file**

Replace the configuration file /opt/ktae/conf/application.conf with the one you backed up from the previous Kaspersky Threat Attribution Engine installation (see the *Backup of Kaspersky Threat Attribution Engine files* step).

9 **Starting Kaspersky Threat Attribution Engine service**

Execute the script to start the Kaspersky Threat Attribution Engine service:

```
# sudo bash 02_run_ktae.sh
```

Adding a custom certificate

This section explains how to configure Kaspersky Threat Attribution Engine to use a custom certificate.

Kaspersky Threat Attribution Engine uses one of the following type of certificates:

- Self-signed certificate for Transport Layer Security (TLS)

  This certificate is generated automatically during Kaspersky Threat Attribution Engine installation and used by default.

- Custom SSL certificate

  This certificate is more secure, and we recommend that you use it. For example, it can be a signed certificate used in your organization.

Only administrators can add a custom certificate to Kaspersky Threat Attribution Engine.

To add a custom certificate to Kaspersky Threat Attribution Engine:

1. Sign a custom certificate in PKCS12 format.
2. Place the certificate in a directory where Kaspersky Threat Attribution Engine will use it.

   By default, Kaspersky Threat Attribution Engine uses a certificate located in the /opt/ktae/cert directory.
3. Go to the /opt/ktae/conf/ directory and open the Kaspersky Threat Attribution Engine configuration file application.conf.
4. In the `tls-settings` section of the file, change the following parameters:

   - Specify the path to the custom certificate in the `key-store-path` parameter.

     If you placed the custom certificate to the default /opt/ktae/cert directory, specify it in this parameter.
   - Specify the password for the custom certificate in the `key-store-password` parameter.

     This password will be required to change or edit the certificate.
5. Grant Kaspersky Threat Attribution Engine users read access to the certificate so that they can add it to a list of trusted certificates in their browser.

Upgrading Kaspersky Threat Attribution Engine

This section explains how to upgrade Kaspersky Threat Attribution Engine from a previous version to the latest version.

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

Scenario for Kaspersky Threat Attribution Engine 2.x upgrade to the 2.3 version is described in the "*Migration of existing instance from CentOS 7 to Rocky Linux 9.4*" section.

Preparing for upgrading

Before upgrading the Kaspersky Threat Attribution Engine, do the following:

1. Create the ktae_updater user if it does not exist yet:
   ```
   sudo adduser -s /bin/bash ktae_updater
   sudo passwd ktae_updater
   ```
2. If necessary, run the following command to give the ktae user access to bash:

```
sudo chsh -s /bin/bash ktae
```

3. Make sure the ktae user is the owner of the /opt/ktae directory. Also make sure that it has access to the /opt/ktae-updater directory and the user ktae_updater is the owner.

   If necessary, run the following commands to change the owner of the directories to the ktae user:

   ```
   sudo chown -R ktae:ktae /opt/ktae
   sudo chown -R ktae_updater:ktae_updater /opt/ktae-updater
   ```

Upgrading the Updater service

For both configurations (simple or air-gapped), the Updater service should be upgraded in the open perimeter.

To upgrade the Updater service:

1. Run the following command to start the upgrade:

   ```
   cd distr/upgrade; sudo bash upgrade_ktae_updater.sh
   ```

   In the command above, distr is the directory with the latest version of the Kaspersky Threat Attribution Engine distributive.

2. In the menu that appears in the terminal, select the `Install KTAE updater` option.

   The Updater service will be upgraded to the latest version. The console will display the version of the Updater service.

Upgrading the Kaspersky Threat Attribution Engine service

Depending on the configuration you selected to install Kaspersky Threat Attribution Engine, you should upgrade the Kaspersky Threat Attribution Engine service in different perimeters:

- For the simple configuration, run the steps in the open perimeter.
- For the air-gapped configuration, run the steps in the separate secure perimeter.

To upgrade the Kaspersky Threat Attribution Engine service:

1. Back up the Kaspersky Threat Attribution Engine database.
2. Delete all files from the `<SECURE_UPDATES_DIR>`/bases directory.

   The `<SECURE_UPDATES_DIR>` was created in the secure perimeter when preparing to install Kaspersky Threat Attribution Engine.

3. Launch the admin_cli utility.
4. Run the following command to stop the Kaspersky Threat Attribution Engine service:

   ```
   terminate-service
   ```

5. Exit the admin_cli utility as follows, or launch a new terminal session.

   ```
   quit
   ```

6. Run the upgrade script:

   ```
   cd distr/upgrade; sudo bash upgrade_ktae.sh
   ```

In the command above, distr is the directory with the latest version of the Kaspersky Threat Attribution Engine distributive.

7. In the menu that appears in the terminal, select the `Install KTAE service` option.

   The Kaspersky Threat Attribution Engine service will be upgraded.

   > The script will create a backup for the configuration files of the previous Kaspersky Threat Attribution Engine service version in the /opt/ktae/config_backup directory.

8. Return to the terminal session with the admin_cli utility, or launch the utility again.

9. Run the following command to start the Kaspersky Threat Attribution Engine service:

   ```
   run-service
   ```

Uninstalling Kaspersky Threat Attribution Engine

This section explains how to uninstall Kaspersky Threat Attribution Engine services.

Uninstalling the Updater service

For both configurations, the Updater service is uninstalled from the open perimeter.

To uninstall the Updater service:

1. Run the following command to delete the Updater RPM package:

   ```
   sudo rpm -e ktae-updater
   ```

2. Delete the directory where the Updater was installed.

   By default, this directory is /opt/ktae-updater.

Uninstalling the Kaspersky Threat Attribution Engine service

Depending on the configuration you used to install Kaspersky Threat Attribution Engine, run the uninstallation steps in different perimeters:

- For the simple configuration, run the steps below in the open perimeter.
- For the air-gapped configuration, run the steps below in the separate secure perimeter.

To uninstall the Kaspersky Threat Attribution Engine service:

1. Run the following command to stop the Kaspersky Threat Attribution Engine service:

   ```
   systemctl stop ktae
   ```

2. Run the following command to delete the Kaspersky Threat Attribution Engine RPM package:

   ```
   sudo rpm -e ktae-bin
   ```

3. Run the following command to connect to MySQL:

   ```
   mysql -u root –p
   ```

4. When asked, enter the password for the root user in MySQL.

5. Run the following command to delete the Kaspersky Threat Attribution Engine database:

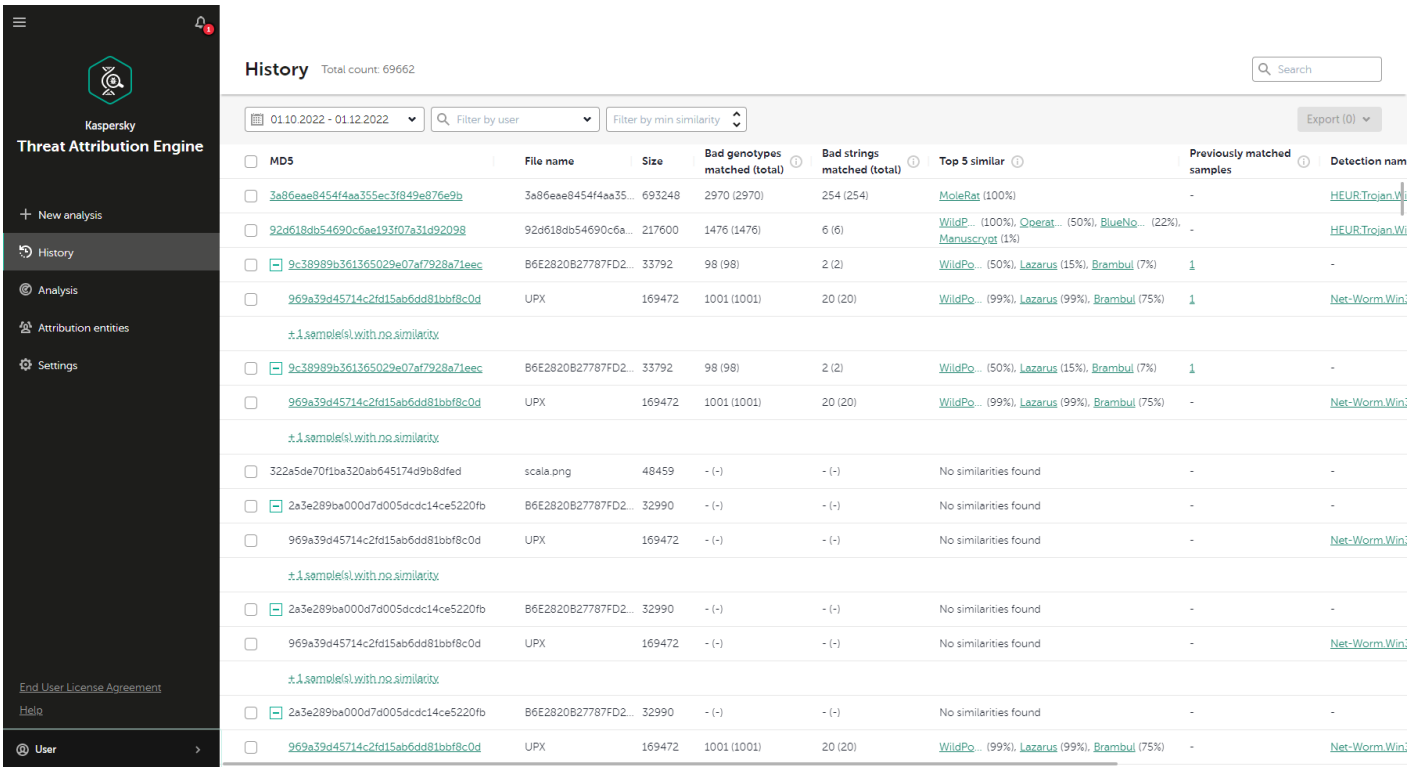
   ```
   drop database ktae;
   ```

6. Delete the directory where Kaspersky Threat Attribution Engine was installed to delete the remaining files. By default, this directory is /opt/ktae.

You successfully uninstalled the Kaspersky Threat Attribution Engine and its services.

Kaspersky Threat Attribution Engine web interface

This section describes the primary elements of the application interface.



Kaspersky Threat Attribution Engine web interface

The table below describes the main elements of the Kaspersky Threat Attribution Engine web interface.

Kaspersky Threat Attribution Engine interface elements

Element	Location	Description	Element content
Main menu	On the left	Provides access to the main functions of the application.	• Menu icon —Collapse or expand the menu. • Notification icon —Show notifications. The button has a badge showing the number of received notifications. • **New analysis**—Opens the **New analysis** window to send a file for analysis and attribution. • **History**—Opens the **History** section to see previously submitted analyses. • **Analysis**—Open the **Analysis** section to see the latest analysis result. • **Attribution entities**—Opens the **Attribution entities** section to manage attribution entities that are known to Kaspersky Threat Attribution Engine. • **Settings**—Opens the **Settings** section to see information about Kaspersky Threat Attribution Engine and the current license. Users with administrator rights can also configure Kaspersky Threat Attribution Engine parameters and resolve update conflicts. • Account icon —Shows the user name of the current user. Click the user name to open account actions. • **End User License Agreement**—Opens the End User License Agreement. • **Help**—Opens the Kaspersky Threat Attribution Engine help in a separate window.
Working window	On the right	Shows content and provides access to functionality of the	The window elements depend on the section you open. Use the main menu to switch between sections and access Kaspersky Threat Attribution Engine functionality.

		section you are currently working with.	

Licensing

This section covers the main aspects of application licensing.

About the End User License Agreement

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

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

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

You can view the terms of the End User License Agreement as one of the following:

- In the main menu, click the **End User License Agreement** link (in the expanded menu) or the End User License Agreement icon (in the collapsed menu).
- Open the license_KTAE_2_3_0_en.rtf file in the /opt/ktae/docs directory.

About licensing

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

A license entitles you to the following kinds of services:

- Using Kaspersky Threat Attribution Engine in accordance with the terms of the End User License Agreement
- Getting technical support

The license can have the following limitations:

- Each license has an expiration date.

  When a license expires, you can no longer perform analysis and attach samples to attribution entities.

- Some license types have a weekly limit on a maximum number of analyzed files.

  When you exceed your weekly analysis limit, you can no longer perform analysis and attach samples to attribution entities.

  The limit is reduced on each analysis you run. When you send a file for analysis, the file is unpacked (if required) and Kaspersky Threat Attribution Engine calculates how many units from the analysis limit will be spent for this analysis. If you have available analysis units, Kaspersky Threat Attribution Engine displays the number of files that will be sent for analysis. This can help you decide whether to analyze the files and spend the analysis units or cancel the analysis. If the number of analysis units required exceeds the number of available analysis units, the process is interrupted and you get the corresponding notification.

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

About the license key

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

Kaspersky Threat Attribution Engine components use different licensing processes:

- License key and license information for Kaspersky Threat Attribution Engine is stored in a USB token.

  This token is required to install Kaspersky Threat Attribution Engine and work with it.

- License key for Kaspersky Threat Attribution Engine updater service is provided as a KEY file with the application distribution pack.

  This license key is required for downloading and updating Kaspersky Threat Attribution Engine and updating anti-virus databases using the Kaspersky Threat Attribution Engine updater service.

License key statuses

Kaspersky Threat Attribution Engine periodically checks the license key status stored in the USB token. Depending on the license key status, Kaspersky Threat Attribution Engine determines if the application can be used under the license. The license key status can be one of the following:

- *License key is valid.*

  You can perform analysis and attach samples to attribution entities. Kaspersky Threat Attribution Engine will be updated.

- *You do not have a license key.*

  This might be because license key information is missing. You cannot perform analysis and attach samples to attribution entities. Kaspersky Threat Attribution Engine will not be updated. Please contact Technical Support.

- *License key has expired.*

  You cannot perform analysis and attach samples to attribution entities. Kaspersky Threat Attribution Engine will not be updated. You must get a new license key. Please contact Technical Support.

- *License key is temporarily blocked.*

  This might be because you have incorrect date and time set on a computer. See the Removing temporarily blocked status section below for instructions on how to fix this status.

- *License key is blocked.*

  You cannot use the license key stored in the USB token because of one of the following reasons:

  - You have exceeded a weekly analysis quota for your license. Wait for the quota to reset.

  - License key is invalid or an error happened. You might need to obtain a new USB token. Please contact Technical Support.

Removing temporarily blocked status

To remove a temporarily blocked status from a license key:

1. On a computer where Kaspersky Threat Attribution Engine is installed, set the correct date and time.
2. Restart Kaspersky Threat Attribution Engine.

Viewing your current license

To view information about your current license,

In the main menu, click **Settings**, and then open the **License** tab.

The **License** tab displays information about the current license that is active on your account. This information is stored on a USB token and in the Kaspersky Threat Attribution Engine database.

The table below describes information shown in the tab.

License information

Field	Description
License key type	Type of the license key that is used on your account.
License key expires on	Date and time when the license key expires.
Days until expiration	Days left until the license key expires. This field is shown only if your license key is active.
License key has expired	Days since the license key has expired. This field is shown only if your license key has expired.
Maximum analyses per week	Number of analyses available until the end of the current week and a total weekly analysis limit that are available for your license. This information is displayed only if your type of license key has this limit.

If your license has expired, the tab displays the corresponding notification.

Data provision

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

When you use Kaspersky Threat Attribution Engine, your data is processed as describes below.

Data sent outside the customer infrastructure perimeter

Kaspersky Threat Attribution Engine sends the following data outside the customer infrastructure perimeter:

- Identifier of the Kaspersky Threat Attribution Engine application
- License identifier
- Update session identifier
- Full version of the Kaspersky Threat Attribution Engine update utility: major, minor, build, revision, and hotfix

Sending the data is required to download the updates. The data is transmitted automatically over a secure channel when the Kaspersky Threat Attribution Engine updates are downloaded from Kaspersky update servers.

No other data is sent outside the customer infrastructure perimeter.

Data stored by Kaspersky Threat Attribution Engine

Kaspersky Threat Attribution Engine stores data that users provide to educate the application.

The table below describes what data is stored in the user infrastructure. No data is transferred to Kaspersky infrastructure.

Data stored by Kaspersky Threat Attribution Engine

Storage location	Data	Storage period	Access control
Log files located at the Kaspersky Threat Attribution Engine local server	• User identifier, user name, connection identifier, session identifier • File fragments (extracted genotypes and strings), analysis and sample attachment request metadata Metadata is MD5 hash of an analyzed file, file name, size of extracted strings and genotypes, and metadata of attribution entities.	30 days or up to 1 gigabyte (GB)	Linux® file system security
MySQL database located at the Kaspersky Threat Attribution Engine local server	• User identifier, connection identifier • User name, user login, user role, hash, and a password salt • MD5 hashes of analyzed files, file names, file sizes, file signatures, extracted paths (for archived or packed files), user identifier, connection identifier, analysis date, and file fragments (extracted genotypes and strings)	Unlimited	MySQL security
User home directory located at the Kaspersky Threat Attribution Engine local server	• File confirming that user accepted the End User License Agreement terms during the Kaspersky Threat Attribution Engine installation • Metadata contained in the downloaded report about the Kaspersky Threat Attribution Engine analysis results Metadata is MD5 hash of an analyzed file, file name, file size, file signature, extracted path (for archived or packed files), user name, analysis date, file fragments (extracted genotypes and strings), attribution entity information.	Unlimited	Linux file system security

Temporary storage located at the Kaspersky Threat Attribution Engine local server	• Files sent for analysis and sample attachment • File fragments (genotypes and strings)	Until the end of task processing	Linux file system security
Browser cookie storage	Session identifier	Unlimited	Browser security

Getting started with Kaspersky Threat Attribution Engine

This section describes how to start using Kaspersky Threat Attribution Engine web interface.

Before signing in, obtain the following from your Kaspersky Threat Attribution Engine administrator:

- IP address or domain for Kaspersky Threat Attribution Engine web interface
- Login
- Password

Adding a certificate (recommended)

To access to Kaspersky Threat Attribution Engine, we recommend that you add a certificate to a list of trusted certificates in your browser. By default, Kaspersky Threat Attribution Engine uses a self-signed certificate for Transport Layer Security (TLS), which is generated automatically during Kaspersky Threat Attribution Engine installation. An administrator can also configure Kaspersky Threat Attribution Engine to work with a custom certificate.

> You must add a certificate to the list of trusted certificates only once.

To add the Kaspersky Threat Attribution Engine certificate:

1. In your browser, navigate to `https://<KTAE_IP_ADDRESS>:8080` or `https://<KTAE_DOMAIN>:8080`.

   In the web addresses above, replace `<KTAE_IP_ADDRESS>` with the obtained IP address and `<KTAE_DOMAIN>` with the obtained domain.

2. Add the certificate to the list of trusted certificates using instructions for your browser.

Signing in to Kaspersky Threat Attribution Engine

To sign in to Kaspersky Threat Attribution Engine:

1. In your browser, navigate to `https://<KTAE_IP_ADDRESS>:8080` or `https://<KTAE_DOMAIN>:8080`.

   Replace `<KTAE_IP_ADDRESS>` with the obtained IP address and `<KTAE_DOMAIN>` with the obtained domain.

   > If you already added the Kaspersky Threat Attribution Engine certificate to the list of trusted certificates, but you see a privacy error window, you must stop working with Kaspersky Threat Attribution Engine and contact the administrator. This might indicate a security issue.

2. On the **Sign in** page, enter the login and password you obtained from your administrator.
3. Click **Sign in**.

After you successfully sign in to Kaspersky Threat Attribution Engine, you can send files for analysis and attribution, view analysis results, manage attribution entities.

Signing out of Kaspersky Threat Attribution Engine

To sign out of Kaspersky Threat Attribution Engine,

In the lower part of the main menu, click your user name, and then click **Log out**.

You will be redirected to the **Sign in** page.

Analyzing files in Kaspersky Threat Attribution Engine

This section explains what file types Kaspersky Threat Attribution Engine can analyze, how license analysis limit is spent, and how to send a file for analysis and attribution.

About analyzed files

This section describes what file types you can send for analysis and how Kaspersky Threat Attribution Engine operates with the files.

About supported file types

You can send the following file types for analysis:

- **Individual files**

  You can send individual files for analysis if they have the following characteristics:

  - The file can be of any type, such as an executable file, Portable Document Format (PDF) files, Microsoft Word documents, Java scripts, or certificate files.
  - Size of each individual file must be from 6 bytes to 500 megabytes (MB).
  - If you send several files is a batch, total number of files in the batch must be up to 100 and total size of all the files must be up to 500 MB.

- **Archives**

  Kaspersky Threat Attribution Engine can unpack the archives you send for analysis. You can send one archive or several archives in a batch (up to 100) for analysis. The archive or each archive in a batch must have the following characteristics:

  - Each file in the archive must be from 6 bytes to 500 MB in size after it is unpacked.
  - One archive must contain up to 1000 files. If the file contains more than 1000 files or if the unpacking timeout is exceeded, a warning is displayed in the analysis results.
  - Archive can be protected with a password.

- **Memory dump files**

  Kaspersky Threat Attribution Engine can analyze raw memory dump files and look for similarities with known attribution entity samples. For example, you can create a memory dump file on a system that might be compromised and analyze it in Kaspersky Threat Attribution Engine to identify any malicious code fragments. This can be useful during incident response or when a system is infected with fileless malware.

  You can send one or several memory dump files in a batch (up to 100) for analysis using one of the following methods:

  - Upload the memory dump file in blocks. Each block must be up to 500 MB in size.
  - Extract the running processes from the memory dump file as separate files, using an external tool, such as Volatility, and upload the files separately.

    You can use Volatility to extract the processes as follows:

```
volatility-2.2.standalone.exe -f %memdump.vmem% procexedump --dump-
dir=%dumped_files_dir%
```

In the command above, replace `%memdump.vmem%` with the name of the memory dump file, `%dumped_files_dir%` with a path to a directory where the resulting files must be stored.

About samples

A *sample* is a file with which Kaspersky Threat Attribution Engine operates. Sample can be sent for analysis to Kaspersky Threat Attribution Engine or attached to an attribution entity. If a sample is an archive, individual samples unpacked from it are called *nested samples*. Kaspersky Threat Attribution Engine analyzes each nested sample individually.

Kaspersky Threat Attribution Engine can receive samples as one of the following:

- *Trusted samples*—Samples that are determined as clean and not attributed to any entity.

  These samples are provided by Kaspersky experts with Kaspersky Threat Attribution Engine database updates.

- *Previously analyzed samples*—Samples that are sent for analysis and attribution by Kaspersky Threat Attribution Engine users.

- *Attribution entity samples*—Samples that are attached to an attribution entity. *Attribution entity* can be a known advanced persistent threat (APT) actor, its tool, related malware, or campaign that were previously used in cyberattacks.

  Kaspersky Threat Attribution Engine can receive new attribution entity samples as one of the following:

  - Provided by Kaspersky experts with Kaspersky Threat Attribution Engine database updates.

  - Attached to an attribution entity by users (*custom* attribution entity samples).

From each received sample, Kaspersky Threat Attribution Engine automatically extracts *genotypes* (distinctive pieces of binary code) and *strings* (distinctive strings of characters) and uses them for comparison and attribution. Genotypes and strings extracted from the trusted samples are considered *trusted* and can be used to determine a new sample as clean.

Depending on similarity between the analyzed sample and attribution entity samples in the Kaspersky Threat Attribution Engine database, the analyzed sample can be determined as one of the following:

- Attributed to an attribution entity if genotypes and strings of the analyzed sample are similar to genotypes and strings in the already attached samples.

  In this case, similar genotypes and strings are called *bad genotypes* and *bad strings*.

- Clean and not attributed to any attribution entity if genotypes and strings of the analyzed sample are similar to genotypes and strings in trusted samples.

- Not categorized if the analyzed sample has no genotypes and strings that are similar to attribution entity samples or the number of similar strings and genotypes is less than the string threshold and genotype threshold.

Each received and analyzed sample is added to the Kaspersky Threat Attribution Engine database with its metadata, such as MD5 hash, extracted genotypes and strings, file size, and attribution to entities.

About license analysis limit

This section explains how license analysis limit is spent when you send files for analysis.

Your license can have a limitation on a maximum number of files you can analyze per week. The limit is reduced on each analysis you run. When you send a file for analysis, Kaspersky Threat Attribution Engine calculates how many units from the analysis limit will be spent for this analysis. Depending on the file you send for analysis, the analysis limit is spent as follows:

- If you send an individual compound object for analysis (for example, a PDF or Microsoft Word document), the weekly limit decreases only by one unit, and not by the total number of unique files extracted from this file. It means that if you send a wrong file for analysis, the analysis limit does not decrease critically.
- If you send an archive file for analysis, it is unpacked (if required), and Kaspersky Threat Attribution Engine calculates the quota units that will be spent for this analysis. Each unpacked file decreases the weekly limit.
- If you send the same file for analysis, the weekly analysis limit will be reduced again. This happens because the analysis limit is reduced on each analysis.
- If you cancel an analysis while its running, the weekly analysis limit will be reset to the same number as it was before the analysis.

When you send files to Kaspersky Threat Attribution Engine, it displays the number of files that will be analyzed. This can help you decide whether to analyze the files and spend the analysis units or cancel the analysis. If the number of analysis units required exceeds the number of available analysis units, the process is interrupted and you get the corresponding notification.

Sending files for analysis and attribution

To send files for analysis and attribution:

1. Open the **New analysis** window as one of the following:
   - In the main menu, click the **New analysis** button.
   - In the main menu, click **Analysis** and click the **New analysis** button in the upper part of the page.
2. Add files for analysis and attribution as one of the following:
   - Drag and drop one or more files into the selection zone.
   - Click the **Select files** button and select one or more files to add.

   See About supported file types for information about file types you can send for analysis.

   > Your account must have read permissions for the files. Otherwise, your browser will not be able to upload it.

3. If you want to ignore similarity thresholds for genotypes and strings that will be extracted from the files, select **Reset similarity thresholds**.

   If the check box is selected, file samples are considered to be similar during analysis if they have at least one matching genotype or string.

4. If you want to unpack the file and analyze its contents as well, select **Unpack**.

5. If you select the **Unpack** check box to unpack the file and the file is protected with a password, specify the password in the **Unpack password** field.

   Maximum password length is 256 symbols. If you do not specify the password, Kaspersky Threat Attribution Engine will use several default passwords, such as `infected`, `virus`, `malware`, `password`, `qwerty`, to try unpacking the file.

   > For a complete list of default passwords, see a file located at /opt/ktae/tools/linux/scanner/passwords. Each line in the file (including blank lines) is a possible password.

6. Click **Start** to send the files for analysis.

7. If your license has a weekly analysis limit, click **Continue** in the confirmation window to confirm sending the files for analysis, or click **Cancel** to discard sending the files.

   Kaspersky Threat Attribution Engine calculates the number of analysis units that are required to analyze the files. If you have enough units available, the following information is displayed in the confirmation window:

   - Total number of files that will be sent for analysis
   - Number of files extracted from the file you sent
   - Number of files from which other files were extracted

   If you do not have enough units available to perform the analysis, the corresponding message is displayed. See About license analysis limit for information about how the license limit is spent.

8. When the analysis completes, view the results displayed in the **Analysis** section.

Working with analysis results

This section explains how to view analysis statuses and results for the samples you sent for analysis and attribution.

About analysis statuses

After you send a file for analysis and attribution, you are redirected to the **Analysis** section that shows recent analysis results. The **Top 5 similar** column shows a list of attribution entities with similar samples or one the analysis statuses listed below if no attribution entities are found.

This information is also shown in the **Top 5 similar** column of the **History** section.

The following statuses can be shown:

- *Canceled*—Analysis was canceled by a user.
- *Sample is trusted*—The analyzed sample is trusted, which means that it is associated with a file that is marked as *clean* or *trusted* by Kaspersky.

  A list of trusted samples, genotypes, and strings is provided with Kaspersky Threat Attribution Engine database updates. Kaspersky Threat Attribution Engine users cannot assign the *trusted* status to samples, genotypes, or strings.
- *File not found*—An error occurred because the file sample sent for analysis was not found on the server.
- *Empty extraction*—No strings or genotypes were extracted from the sample.
- *No similarities found*—No known attribution entity samples match the analyzed sample.
- *Extraction has failed*—An error occurred while extracting strings and genotypes from the sample.
- *Size is not valid*—The analyzed sample exceeds a size limit. The size of the analyzed sample must be from 6 bytes to 500 MB.
- *Duplicate*—The sample has duplicates in the same analysis.
- *Timeout exceeded*—Analysis operation exceeded timeout.
- *Internal error*—An error occurred while analyzing the sample.

If you sent a sample for unpacking during the analysis, the **Analysis** and **History** sections also show the sample unpacking status. To view this status, hover over an icon near the name of the analyzed sample in the **File name** column. The following statuses can be shown:

- No status if unpacking was successful
- *Unpacking operation has failed*
- *Not all unpacked samples were processed because the unpacked file limit was reached*
- *Sample is protected with an unknown password*

- *Sample is corrupted*

Viewing recent analysis result

This section explains how to view result for a recently submitted analysis and what information you can view on the **Analysis** section.

After you send a file for analysis and attribution, the **Analysis** section opens automatically. You can also open the section by clicking **Analysis** in the main menu.

This section shows the following information:

- Status of the currently running analysis if you have an active analysis in progress.

  You can cancel the running analysis by clicking **Cancel**.

- Analysis result table that shows result summary for the last analysis you submitted.

  You can click a table row to open a right-side pane displaying the analysis result for the corresponding sample only.

In the **Analysis** section, you can also start a new analysis, search analysis data, and export analysis results.

If the sample was unpacked during analysis, individual samples extracted from it can also be shown in the result table as nested samples. In this case, only the following nested samples are displayed:

- Samples that match known attribution entities
- Samples that match previously analyzed samples
- Samples that have detection names
- Samples that were unpacked with an error

All other nested samples are hidden. To show all nested samples, if any, you can click the **+ N sample(s) with no similarity** or the expand icon ⊞ near the parent sample.

The table below describes information that is shown about the analyzed sample in the result table and the right-side pane.

Analysis result table

Table field	Description	Action
MD5	MD5 hash of the analyzed file.	If Kaspersky Threat Attribution Engine found a similar sample, you can click the MD5 hash and open a detailed analysis report for the analyzed sample.
File name	Name of the analyzed sample.	None.
Size	Size of the analyzed sample, in bytes.	None.
Bad genotypes matched (total)	Number of genotypes in the analyzed sample that matched genotypes in similar attribution entity samples. The number in parenthesis indicates the total number of matching genotypes in similar attribution entity samples if the similarity threshold is ignored.	None.
Bad strings matched (total)	Number of strings in the analyzed sample that matched strings in similar attribution entity samples. The number in parenthesis indicates the total number of matching strings in similar attribution entity samples if the similarity threshold is ignored.	None.

Top 5 similar	Top five attribution entities that own the most similar samples. The similarity percentage is shown in parenthesis. If similar attribution entity samples were not found, the field shows an analysis status, for example, indicating if the sample is trusted or if errors happened during the analysis.	You can click an attribution entity name to open attribution entity properties. If the field shows a *Sample is trusted* status, it means that the sample is associated with a file that is marked as clean or trusted by Kaspersky. If the field shows a *No similarities found* status, it means the analyzed samples has no similar samples related to attribution entities.
Previously matched samples	Number of similar samples among the previously analyzed samples.	You can click the number in this column to open the analysis result for previously analyzed samples. If the field does not show a number, it means the analyzed sample has no similar samples among the previously analyzed samples, or the sample was sent for analysis using API and searching among the previously analyzed samples was disabled.
Detection names	Names of detections found in the analyzed sample by the anti-virus engine.	You can click a detection name to go to the Kaspersky Threats portal page for the object and view its detailed description, such as date of the detection discovery, detection class, platform, top 10 attacked countries. If no information about the detect object is available on the portal, the link goes to a description of the detection class. You can also hover over the detection name and click **Copy to clipboard** in the opened pop-up to copy the name for further research in other systems.

Viewing analysis history

This section explains how to view analysis history and what information you can view on the **History** section.

On the **History** section, you can view the analysis history for post-research purposes and determining available license limit, search analysis history, and export analysis results. The history storage period is set to 180 days.

> Administrators can view the history storage period set in the **Storage period (days)** parameter in the **Settings** section. However, this parameter cannot be modified.

Depending on a user role, you might have limited access to analysis history:

- Regular users can only view analyses that they submitted.
- Users with administrator rights can view analyses that all users submitted.

The analysis history is shown as a table with result summary. Each table row shows result for an individual sample. You can click a table row to open a right-side pane displaying the analysis result for the corresponding sample only.

If the sample was unpacked during analysis, individual samples extracted from it can also be shown in the result table as nested samples. In this case, only the following nested samples are displayed:

- Samples that match known attribution entities
- Samples that match previously analyzed samples
- Samples that have detection names
- Samples that were unpacked with an error

All other nested samples are hidden. To show all nested samples, if any, you can click the **+ N sample(s) with no similarity** or the expand icon ⊞ near the parent sample.

If you switch to the **History** section after an analysis is completed, a message *History is not up to date. Refresh page* is shown at the upper part of the page. Click **Refresh page** link of the message to update the analysis history.

The table below describes information that is shown about the analyzed samples in the result table and the right-side pane.

History table

Table field	Description	Action
MD5	MD5 hash of the analyzed file.	If Kaspersky Threat Attribution Engine found a similar sample, you can click the MD5 hash and open a detailed analysis report for the analyzed sample.
File name	Name of the analyzed sample.	None.
Size	Size of the analyzed sample, in bytes.	None.
Bad genotypes matched (total)	Number of genotypes in the analyzed sample that matched genotypes in similar attribution entity samples. The number in parenthesis indicates the total number of matching genotypes in similar attribution entity samples if the similarity threshold is ignored.	None.
Bad strings matched (total)	Number of strings in the analyzed sample that matched strings in similar attribution entity samples. The number in parenthesis indicates the total number of matching strings in similar attribution entity samples if the similarity threshold is ignored.	None.
Top 5 similar	Top five attribution entities that own the most similar samples. The similarity percentage is shown in parenthesis. If similar attribution entity samples were not found, the field shows an analysis status, for example, indicating if the sample is trusted or if errors happened during the analysis.	You can click an attribution entity name to open attribution entity properties. If the field shows a *Sample is trusted* status, it means that the sample is associated with a file that is marked as clean or trusted by Kaspersky. If the field shows a *No similarities found* status, it means the analyzed samples has no similar samples related to attribution entities.
Previously matched samples	Number of similar samples among the previously analyzed samples.	You can click the number in this column to open the analysis result for previously analyzed samples. If the field does not show a number, it means the analyzed sample has no similar samples among the previously analyzed samples, or the sample was sent for analysis using API and searching among the previously analyzed samples was disabled.
Detection names	Names of detections found in the analyzed sample by the anti-virus engine.	You can click a detection name to go to the Kaspersky Threats portal page for the object and view its detailed description, such as date of the detection discovery, detection class, platform, top 10 attacked countries. If no information about the detect object is available on the portal, the link goes to a description of the detection class. You can also hover over the detection name and click **Copy to clipboard** in the opened pop-up to copy the name for further research in other systems.
Date	Date and time when the analysis was performed.	None.
User	Name of a user who performed the analysis.	None.

Exploring detailed analysis report

If Kaspersky Threat Attribution Engine finds samples that are similar to the analyzed sample, a detailed analysis report is available where you can learn more about related attribution entities and matched genotypes and strings.

To open a detailed analysis report:

1. In the main menu, click **Analysis** or **History**.
2. In the **MD5** column, click any underlined sample hash.
   The detailed analysis report for the sample opens.

The report can help you determine the following information:

- Attribution entity (a group or an individual) that performed the attack
- Attribution entity motivation for the attack
- Attribution entity tactics, techniques, and procedures (TTP)

In the detailed analysis report, you can also search analysis results and export result data.

The report consists of a general information section and two tabs showing information about samples that are similar to the analyzed sample.

General information

The general information section in the upper part of the report shows general information about the analyzed sample. You can click the section area to open the sample information in the right-side pane, which shows a full list of matched attribution entities and detected objects.

The table below describes information that is shown about the analyzed sample in the section.

General information section

Field	Description	Action
Size	Size of the analyzed sample, in bytes.	None.
Extracted path	Path to the current sample relative to the parent sample that was sent for analysis. This field is shown only for samples that were unpacked for analysis.	None.
Matched attribution entities	List of attribution entities that matched the analyzed sample with similarity percentage. If the similarity percentage is 100%, it means that this attribution entity owns this sample.	If Kaspersky Threat Attribution Engine finds similarity between the analyzed sample and a sample of a known attribution entity, we recommend that you familiarize yourself with the attribution entity. You can click an attribution entity to open attribution entity properties and examine the available information about it. The section shows the following information: • General information about a related attribution entity, campaign, or malware. • Links to public reports related to the attribution entity. • List of samples attached to the attribution entity.
Detection names	Names of detections found in the analyzed sample by the anti-virus engine.	You can click a detection name, to go to the Kaspersky Threats portal page for the object and view its detailed description, such as date of the detection discovery, detection class, platform, top 10 attacked countries. If no information about the detected object is available on the portal, the link goes to a description of the detection class. You can also hover over the detection name and click **Copy to clipboard** in the opened pop-up to copy the name for further research in other systems.

Attribution entity samples

The **Attribution entity samples** tab in the lower part of the report shows information about attribution entity samples that are similar to the analyzed sample, as well as genotypes and strings that matched in the samples.

If Kaspersky Threat Attribution Engine finds only previously analyzed samples that are similar to the analyzed sample, the **Attribution entity samples** tab is empty.

The information about similar attribution entity samples, matched genotypes, and matched strings is shown in separate tables. For each table, the number in parenthesis indicates the total number of entries in the table. You can click the open report icon near the table title to open the table in a right-side pane and examine the data.

The table below describes information that is shown about the analyzed sample in the section tables.

Attribution entity samples tab

Table name	Description	Table fields
Similar samples	Information about attribution entity samples that are similar to the analyzed sample.	**MD5**—MD5 hash of a similar attribution entity sample. **Size**—Size of a similar attribution entity sample, in bytes. **Matched genotypes**—Numbers of matched and total genotypes in the similar attribution entity sample. The first number indicates how many genotypes in the attribution entity sample match the genotypes in the analyzed sample. The second number indicates the total number of genotypes in the attribution entity sample. **Matched strings**—Number of matched and total strings in the similar attribution entity sample. The first number indicates how many strings in the attribution entity sample match the strings in the analyzed sample. The second number indicates the total number of strings in the similar attribution entity sample. **Similarity**—Similarity percentage between the analyzed sample and the similar attribution entity sample. **Attribution entity**—Attribution entity that owns this similar sample. You can click the attribution entity to open attribution entity properties. In this section, you can examine information about the attribution entity and attach the analyzed sample to the entity. **Aliases**—Known aliases of the attribution entity that owns this similar sample.
Matched genotypes	Information about genotypes that are common in the analyzed sample and a similar attribution entity sample.	**Genotype**—Identifier of a genotype that was extracted from the analyzed sample and matched genotypes in similar attribution entity samples. You can click the genotype identifier to open genotype properties. **Matched**—Number of known attribution entity samples that have this matched genotype. If only one attribution entity has the genotype, it means the genotype is specific to this attribution entity. We recommend that you prioritize investigating such genotypes as they can help you to pinpoint the attack source. Genotypes with many of attribution entities likely belong to a popular malware used by many attribution entities. We do not recommend attributing such genotypes to a specific entity because they cannot be reliably linked to any particular entity. **Used by**—Attribution entities that own samples with this genotype. The number in parentheses specifies how many samples related to the attribution entity have this genotype.
Matched strings	Information about strings that are common in the analyzed sample and a similar attribution entity sample.	**String**—Identifier of a string that was extracted from the analyzed sample and matched strings in similar attribution entity samples. You can click the string identifier to open string properties. **Matched**—Number of known attribution entity samples that have this matched string. If only one attribution entity has the string, it means the string is specific to this attribution entity. We recommend that you prioritize investigating such strings as they can help you to strings the attack source. Strings with many of attribution entities likely belong to a popular malware used by many attribution entities. We do not recommend attributing such strings to a specific entity because they cannot be reliably linked to any particular attribution entity.

		Used by—Attribution entities that own samples with this string. The number in parentheses specifies how many samples related to the attribution entity have this string.

Previously analyzed samples

The **Previously analyzed samples** tab in the lower part of the report shows information about previously analyzed samples that are similar to the analyzed sample, as well as genotypes and strings that matched in the samples.

If Kaspersky Threat Attribution Engine finds only attribution entity that are similar to the analyzed sample, the **Previously analyzed samples** tab is empty.

The information about similar samples, matched genotypes, and matched strings is shown in separate tables. For each table, the number in parenthesis indicates the total number of entries in the table. You can click the open report icon near the table title to open the table in a right-side pane and examine the data.

The table below describes information that is shown in the section tables about the analyzed sample.

Previously analyzed samples tab

Table name	Description	Table fields
Similar samples	Information about previously analyzed samples that are similar to the analyzed sample.	**MD5**—MD5 hash of a similar previously analyzed sample. **File name**—Name of the similar previously analyzed sample. **Size**—Size of the similar previously analyzed sample, in bytes. **Matched genotypes**—Numbers of matched and total genotypes in the similar previously analyzed sample. The first number indicates how many genotypes in the similar previously analyzed sample match the genotypes in the analyzed sample. The second number indicates the total number of genotypes in the similar previously analyzed sample. **Matched strings**—Number of matched and total strings in the similar previously analyzed sample. The first number indicates how many strings in the similar previously analyzed sample match the strings in the analyzed sample. The second number indicates the total number of strings in the similar previously analyzed sample. **Similarity**—Similarity percentage between the analyzed sample and the similar previously analyzed sample. **Top 5 similar**—Top five attribution entities that own the most similar previously analyzed samples. The similarity percentage is shown in parenthesis. You can click the attribution entity name to open attribution entity properties. **Last analysis date**—Date and time when the similar previously analyzed sample was last analyzed. **User**—Name of the user that last analyzed the similar previously analyzed sample.
Matched genotypes	Information about genotypes that are common in the analyzed sample and a similar previously analyzed sample. You can click a row in the table to open a right-side panel with more details about the genotype.	**Genotype**—Identifier of the genotype that was extracted from the analyzed sample and matched genotypes in the similar previously analyzed sample. You can click the genotype identifier to open genotype properties. **Matched**—Number of similar previously analyzed samples that have this matched genotype. **Used by**—Attribution entities that own samples with this genotype. The number in parentheses specifies how many samples related to the attribution entity have this genotype. **File names**—Names of the previously analyzed samples that contain this genotype.
Matched strings	Information about strings that are common in the analyzed sample and a similar previously analyzed sample. You can click a row in the table to open a right-side panel with more details about the string.	**String**—Identifier of the string that was extracted from the analyzed sample and matched strings in the similar previously analyzed sample. You can click the string identifier to open string properties. **Matched**—Number of similar previously analyzed samples that have this matched string. **Used by**—Attribution entities that own samples with this string. The number in parentheses specifies how many samples related to the attribution entity have this string. **File names**—Names of the previously analyzed samples that contain this string.

Viewing genotype properties

This section explains how to open properties for matched genotypes extracted from analyzed samples.

To view genotype properties:

1. In the main menu, click **Analysis** or **History**.
2. In the **MD5** column, click any underlined sample hash.

   The analysis result page for the sample opens.
3. Open the **Attribution entity samples** or **Previously analyzed samples** tab.
4. In the **Matched genotypes** table, click a genotype identifier.

   The page with genotype properties opens.

The page displays a table with information about samples that contain the genotype. In genotype properties, you can also search data and export results.

Depending on the analysis result tab from which you opened the genotype properties, the table shows information about different sample types.

The table below describes information that is shown about genotypes if you opened the page from the **Attribution entity samples** tab.

Genotype properties for attribution entity samples

Table field	Description
MD5	MD5 hash of a similar attribution entity sample with this genotype.
Size	Size of the sample file, in bytes.
Attribution entity	Attribution entity that owns the sample. You can click the attribution entity name to open attribution entity properties.

The table below describes information that is shown about genotypes if you opened the page from the **Previously analyzed samples** tab.

Genotype properties for similar previously analyzed samples

Table field	Description
MD5	MD5 hash of a similar previously analyzed sample with this genotype.
Size	Size of the sample file, in bytes.
File name	Name of the previously analyzed sample that contains this genotype.
Top 5 similar	Top five attribution entities owning samples that are the most similar to the previously analyzed sample. The similarity percentage is shown in parenthesis. You can click the attribution entity name to open attribution entity properties.
Last analysis date	Date and time when the sample was last analyzed.

Viewing string properties

This section explains how to open properties for matched strings extracted from analyzed samples.

To view string properties:

1. In the main menu, click **Analysis** or **History**.
2. In the **MD5** column, click any underlined sample hash.

   The analysis result page for the sample opens.
3. Open the **Attribution entity samples** or **Previously analyzed samples** tab.
4. In the **Matched string** table, click a string identifier.

   The page with string properties opens. The page displays a table with information about samples that contain the string. You can also search data and export results.

Depending on the analysis result tab from which you opened the string properties, the table shows information about different sample types.

The table below describes information that is shown about strings if you opened the page from the **Attribution entity samples** tab.

String properties for attribution entity samples

Table field	Description
MD5	MD5 hash of a similar attribution entity sample with this string.
Size	Size of the sample file, in bytes.
Attribution entity	Attribution entity that owns the sample. You can click the attribution entity name to open attribution entity properties.

The table below describes information that is shown about strings if you opened the page from the **Previously analyzed samples** tab.

String properties for similar previously analyzed samples

Table field	Description
MD5	MD5 hash of a similar previously analyzed sample with this string.
Size	Size of the sample file, in bytes.
File name	Name of the previously analyzed sample that contains this string.
Top 5 similar	Top five attribution entities owning samples that are the most similar to the previously analyzed sample. The similarity percentage is shown in parenthesis. You can click the attribution entity name to open attribution entity properties.
Last analysis date	Date and time when the sample was last analyzed.

Managing attribution entities

This section explains how to manage attribution entities, their properties, and samples that are attached to the attribution entities.

Kaspersky Threat Attribution Engine has two types of attribution entities:

- *Kaspersky attribution entities*—Attribution entity data that is provided by Kaspersky.

  You can add comments and attach custom samples to the attribution entities, but you cannot modify their properties or delete existing attribution entities.

- *Custom attribution entities*—Attribution entity data that is provided by users.

  You can modify attribution entity properties, such as descriptions and comments, and attach samples. You can also create new attribution entities or delete existing attribution entities.

Viewing attribution entity information

You can view a list of all attribution entities known to Kaspersky Threat Attribution Engine, as well as detailed information and attached samples for each attribution entity.

In the **Attribution entities** section and attribution entity properties, you can also search attribution entity data.

Viewing attribution entity list

To view attribution entity list,

In the main menu, click **Attribution entities**, and then open one of the following tabs depending on the attribution entity type you want to view:

- **Kaspersky**—View information about attribution entities provided by Kaspersky.
- **Custom**—View information about custom attribution entities.

The table below describes information that is shown about attribution entities provided in the **Attribution entities** section.

Attribution entity table

Table field	Description
Name	Attribution entity name. You can click an attribution entity name to view its properties. Detailed information about attribution entity properties is provided in the table below.
Samples	Number of samples that this attribution entity owns. If a sample attachment is in progress, the **Attaching** link is displayed. You can click this link to open the **Attachment** tab and view sample attachment log.
Parent	Parent attribution entity of the current entity. You can click the name to view the parent attribution entity properties.
Description	Attribution entity description.
Aliases	List of known aliases for this attribution entity.
Genotype threshold	Minimum value of genotype threshold among all samples attached to the attribution entity.

String threshold	Minimum value of string threshold among all samples attached to the attribution entity.
Comments	Additional information about the attribution entity.

Viewing attribution entity properties

To view properties for an attribution entity:

1. In the main menu, click **Attribution entities**, and then select one of the tabs depending on the attribution entity type you want to view:
   - **Kaspersky**—View information about attribution entities provided by Kaspersky.
   - **Custom**—View information about custom attribution entities.
2. Click an attribution entity name to open its properties.
3. Open one of the tabs on the page to view specific attribution entity information.

The table below describes information that is shown about attribution entities in different tabs of the page.

Attribution entity properties

Tab	Description	Fields
Info	General properties of the attribution entity.	**Name**—Attribution entity name. **Parent**—Parent attribution entity of the current entity. **Genotype threshold**—Minimum value of genotype threshold among all samples attached to the attribution entity. **String threshold**—Minimum value of string threshold among all samples attached to the attribution entity. **Description**—Attribution entity description. **Aliases**—List of known aliases for this attribution entity. **Report links**—Links to reports associated with this attribution entity. **Comments**—Additional information about the attribution entity.
Samples	Information about samples that are attached to the attribution entity. For Kaspersky attribution entities, you can only view custom samples that were attached by users. You cannot view samples that were attached by Kaspersky experts.	**MD5**—MD5 hash of the sample. **Parent**—MD5 hash of the current sample's parent. **Size**—Size of the sample, in bytes. **Genotype threshold**—Minimum number of matching genotypes. An analyzed sample and an attribution entity sample are considered similar if the number of matching genotypes between them is more than the genotype threshold value. The genotype threshold of the attribution entity sample cannot be lower than the corresponding threshold of an attribution entity. **String threshold**—Minimum number of matching strings. An analyzed sample and an attribution entity sample are considered similar if the number of matching strings between them is more than the string threshold value. The string threshold of the attribution entity sample cannot be lower than the corresponding threshold of an attribution entity. **Attachment date**—Date and time when this sample was attached to the attribution entity, in user time zone.
Attachment	Sample attachment log from the current user (for regular users) or from all users (for users with administrator rights).	**MD5**—MD5 hash of the sample. **Parent**—MD5 hash of the current sample's parent. **File name**—File name of the sample. **Owned by**—Name of another attribution entity that the sample is already attached to, if any. **Status**—Attachment operation status.

User—User that attached the sample to the attribution entity.

Only users with administrator rights can view the **User** field.

For more information about the attribution entity, you can search Kaspersky Threat Intelligence Portal. It provides information about matching actors and related YARA rules.

Adding comments to Kaspersky attribution entities

Kaspersky attribution entities are provided by Kaspersky, so they are added, updated, and deleted only when Kaspersky Threat Attribution Engine databases are updated. You cannot delete Kaspersky attribution entities or modify their properties, but you can add comments to them to specify additional information, if necessary.

To add comments to Kaspersky attribution entities:

1. In the main menu, click **Attribution entities**, and then open the **Kaspersky** tab.
2. Click the radio button next to the attribution entity you want to update.

   You can select only one attribution entity at a time.
3. Click the **Edit** button.
4. In the **Update attribution entity** pane that opens, enter a comment in the **Comments** field.
5. Click **Update** to save the comment.

Managing custom attribution entities

This section explains how to manage custom attribution entities, which are created by users.

Creating a custom attribution entity

We recommend that you create a custom attribution entity for each incident.

To create a custom attribution entity:

1. In the main menu, click **Attribution entities**, and then open the **Custom** tab.
2. Click the **Add** button.
3. In the **New attribution entity** pane that opens in the right side of the page, specify the following parameters for a new custom attribution entity:

   - **Name**—Attribution entity name.

This parameter is required. The maximum name length is 255 symbols.

- **Parent**—Parent attribution entity of the current entity.

  You can only select a parent attribution entity from a drop-down list of existing attribution entities. Parent attribution entity is used for grouping several attribution entities together. For example, if you determine that several attacks are performed by the same actor, you can specify it as a parent attribution entity. If the required parent attribution entity does not exist, you can create an attribution entity for this actor and assign it as a parent for all related attribution entities.

- **Genotype threshold**—Minimum value of genotype threshold among all samples attached to the attribution entity.

  We recommend that you use the default genotype threshold value. Change the genotype threshold only if you have problems with attributing samples to the attribution entity.

- **String threshold**—Minimum value of string threshold among all samples attached to the attribution entity.

  We recommend that you use the default string threshold value. Change the string threshold only if you have problems with attributing samples to the attribution entity.

- **Description**—Attribution entity description.

- **Aliases**—List of known aliases for this attribution entity.

  Aliases are used for storing other known names for the attribution entity. For example, if an attribution entity is an actor and has several known pseudonyms, you can specify them in the **Aliases** field.

- **Report links**—Links to reports and online publications associated with this attribution entity.

- **Comments**—Additional information about the attribution entity.

4. Click **Create**.

If needed, you can attach samples to the created attribution entity.

Updating a custom attribution entity

To update a custom attribution entity:

1. In the main menu, click **Attribution entities**, and then open the **Custom** tab.

2. Click the radio button next to the attribution entity you want to update, and then click the **Edit** button.

   You can select only one attribution entity at a time.

3. In the **Update attribution entity** pane that opens in the right part of the page, change the properties of the attribution entity:

   - **Name**—Attribution entity name.

     The maximum name length is 255 symbols.

   - **Parent**—Parent attribution entity of the current entity.

You can only select a parent attribution entity from a drop-down list of existing attribution entities. Parent attribution entity is used for grouping several attribution entities together. For example, if you determine that several attacks are performed by the same actor, you can specify it as a parent attribution entity. If the required parent attribution entity does not exist, you can create an attribution entity for this actor and assign it as a parent for all related attribution entities

- **Genotype threshold**—Minimum value of genotype threshold among all samples attached to the attribution entity.

  We recommend that you use the default genotype threshold value. Change the genotype threshold only if you have problems with attributing samples to the attribution entity.

- **String threshold**—Minimum value of string threshold among all samples attached to the attribution entity.

  We recommend that you use the default string threshold value. Change the string threshold only if you have problems with attributing samples to the attribution entity.

- **Description**—Attribution entity description.

- **Aliases**—List of known aliases for this attribution entity.

  Aliases are used for storing other known names for the attribution entity. For example, if an attribution entity is an actor and has several known pseudonyms, you can specify them in the **Aliases** field.

- **Report links**—Links to reports and online publications associated with this attribution entity.

- **Comments**—Additional information about the attribution entity.

4. Click **Update**.

If needed, you can update samples associated with the attribution entity.

Deleting a custom attribution entity

To delete a custom attribution entity:

1. In the main menu, click **Attribution entities**, and then open the **Custom** tab.
2. Click the radio button next to the attribution entity you want to update.

   You can select only one attribution entity at a time.
3. Click the **Delete** button.
4. In the **Deleting attribution entity** dialog box, confirm deleting the attribution entity.

Managing samples associated with attribution entities

This section explains how to manage attach samples to Kaspersky and custom attribution entities and manage samples that are attached to them.

Attaching samples to attribution entity

We recommend that you use the following guidelines when attaching samples to attribution entities:

- Attach samples that are uncommon or unique for the attribution entity. Executable files, scripts, and documents are the most effective samples.
- Do not attach popular files and files that belong to a legitimate software. Attaching legitimate software files can result in this software being associated to the related attribution entities. To check if a file is popular or legitimate, use Kaspersky Allowlist program and Kaspersky Threat Intelligence Portal (for example, Kaspersky Threat Intelligence Portal (General Access)) services.

Attaching a new sample to attribution entity

To attach a new sample to an attribution entity:

1. In the main menu, click **Attribution entities**, and then open one of following the tabs depending on the attribution entity type to which you want to attach a sample:
   - **Kaspersky**—Open attribution entities provided by Kaspersky.
   - **Custom**—Open custom attribution entities.
2. Click a name of the attribution entity to which you want to attach a sample.
3. On the page with attribution entity properties, open the **Samples** tab.
4. Click the **Attach samples** button.
5. In the **Attach samples** window, add the files to attach to the attribution entity as one of the following:
   - Drag and drop one or more files into the selection zone.
   - Click the **Select files** button and select the file to add.

   You must have read permissions for the uploaded file. Otherwise, your browser will not be able to upload the file.
6. If you want to unpack the selected file and analyze its contents in addition to the uploaded sample, select **Unpack**.
7. If you select the **Unpack** check box to unpack the file and the file is password protected, specify the password in the **Unpack password** field.

   Maximum password length is 256 symbols. If you do not specify the password, Kaspersky Threat Attribution Engine will use several default passwords to try unpacking the file.

   For a complete list of default passwords, see a file located at /opt/ktae/tools/linux/scanner/passwords. Each line in the file (including blank lines) is a possible password.
8. Click **Attach**.

The sample will be uploaded and attached to the selected attribution entity. You can view the attachment status in the **Attachment** tab.

Attaching samples to another attribution entity

To attach an existing sample to another attribution entity:

1. In the main menu, click **Attribution entities**, and then open one of following the tabs depending on the attribution entity type from which you want to detach a sample:
   - **Kaspersky**—Open attribution entities provided by Kaspersky.
   - **Custom**—Open custom attribution entities.
2. Click a name of the attribution entity from which you want to detach a sample.
3. On the page with attribution entity properties, select the **Samples** tab.
4. Click the radio button next to the sample you want to modify.

   You can only select one sample at a time.
5. Click the **Change attribution entity** button above the sample table.
6. In the **Change attribution entity** pane that opens on the right side, select a new attribution entity to attach the sample using the drop-down list.
7. Click **Change**.

   If the sample has a lower similarity threshold than the minimum threshold set for the new attribution entity, Kaspersky Threat Attribution Engine updates the sample similarity threshold with the minimum threshold set for the attribution entity.

Sample attachment statuses

Kaspersky Threat Attribution Engine displays a status of the sample attachment operation in the in the **Attachment** tab. The following statuses are possible:

- *Canceled*—Attachment operation was canceled by a user.
- *Sample is attached*—The sample is successfully attached to the attribution entity.
- *Sample attachment has failed*—An error occurred while attaching the sample.
- *Sample is trusted*—The sample was not attached because it is a trusted sample.
- *File not found*—An error occurred. The file sent for attachment was not found on the server.
- *Empty extraction*—No strings or genotypes were extracted from the sample.
- *Extraction has failed*—An error occurred while extracting strings and genotypes from the sample.
- *Sample is being attached by another user*—Another user already submitted this samples for attaching to the attribution entity, and the attachment is in progress.
- *Sample is owned by an attribution entity*—The sample is already owned by another attribution entity.

- *Duplicate*—A sample with the same MD5 hash was already processed during attachment request.
- *Sample has only trusted strings and genotypes*—All genotypes and strings extracted from the sample are trusted.
- *Size is not valid*—The attached sample exceeds a size limit.
- *Attribution entity not found*—The attribution entity to which you wanted to attach the sample was deleted.
- *Timeout exceeded*—Sample attachment operation timeout exceeded.
- *Internal error*—An internal error occurred while attaching the sample.

The following statuses are possible for samples that are unpacked:

- No status if unpacking was successful
- *Unpacking operation has failed*
- *Not all unpacked samples were processed because the unpacked file limit was reached*
- *Sample is protected with an unknown password*
- *Sample is corrupted*

Editing similarity thresholds

Genotype threshold and *string threshold* values define the minimum number of matching genotypes and strings. An analyzed sample and an attribution entity sample are considered similar if the number of matching genotypes or strings between them is more than the corresponding threshold value.

To edit similarity thresholds for the sample:

1. In the main menu, click **Attribution entities**, and then open one of the following tabs depending on the attribution entity type that you want to see samples for:
   - **Kaspersky**—Open attribution entities provided by Kaspersky.
   - **Custom**—Open custom attribution entities.
2. Click an attribution entity name.
3. On the page with attribution entity properties, open the **Samples** tab.

   For Kaspersky attribution entities, you can only view and edit custom samples that were attached by users. You cannot view samples that were attached by Kaspersky experts.

4. Click the radio button next to the sample you want to modify.

   You can select only one sample at a time.

5. Click the **Edit thresholds** button.

6. In the **Edit similarity thresholds** pane that opens in the right part of the page, enter new values for similarity thresholds:

   - Enter the new genotype threshold value to the **Genotype threshold** field.
   - Enter the new string threshold value to the **String threshold** field.

   The similarity thresholds of a sample cannot be lower than the minimum similarity threshold set in a corresponding attribution entity.

7. Click **Change**.

Deleting samples from attribution entity

To delete a sample from an attribution entity:

1. In the main menu, click **Attribution entities**, and then open one of following the tabs depending on the attribution entity type you want to see samples for:

   - **Kaspersky**—Open attribution entities provided by Kaspersky.
   - **Custom**—Open custom attribution entities.

2. Click an attribution entity name.
3. On the page with attribution entity properties, open the **Samples** tab.

   For Kaspersky attribution entities, you can only view and delete custom samples that were attached by users. You cannot view samples that were attached by Kaspersky experts.

4. Click the radio button next to the MD5 hash of the sample you want to delete.

   You can select only one sample at a time.

5. Click the **Delete** button.
6. In the **Delete sample** dialog box, confirm deleting the sample.

Searching analysis results

This section explains how to search analysis result data in different Kaspersky Threat Attribution Engine sections.

Searching recent analyses

To search recent analysis results,

In the main menu, click **Analysis**, and then enter one or more words to search in the **Search** field above the result table.

Kaspersky Threat Attribution Engine searches the **MD5**, **File name**, **Top 5 similar**, and **Detection names** columns for the entered words. The table is updated automatically to show matching entries.

Searching analysis history

You can search analysis history using filters or by specifying search keywords to find matching entries. You can filter analysis history by analysis date, by user that submitted the analysis, or by minimum similarity percentage.

Filtering by user is available only to users with administrator rights. Regular users do not have this filter.

To filter analysis history by date:

1. In the main menu, click **History**, and then click the **Filter by date** field above the table.
2. In the date filter pane, select time period for which you want to see the submitted analyses as one of the following:
   - Click one of the predefined time filters in the left side of the pane: **Week**, **Two weeks**, **Month**, **Six months**, **Year**.
   - Select a custom time range by selecting a start date in the left-side calendar and an end date in the right-side calendar.
3. Click **Apply**.

   The history table is updated to show only samples that were submitted for analysis during the specified time period.

To filter analysis history by user:

1. In the main menu, click **History**, and then click the **Filter by user** field above the table.
2. Select one or more user names from the drop-down list using check boxes.
3. Click **Apply**.

The history table is updated to show only samples that were submitted for analysis by the selected users.

To filter analysis history by minimum similarity percentage,

In the main menu, click **History**, and then enter a similarity percentage value in the **Filter by min similarity** field above the history table.

You can also use arrows in the field to select the value.

The history table is updated to show only samples with attribution entities (the **Top 5 similar** column) with the similarity percentage equal to or higher than the specified minimum similarity.

To search analysis history using search keywords,

In the main menu, click **History**, and then enter one or more words you want to search in the **Search** field above the table.

Kaspersky Threat Attribution Engine searches the **MD5**, **File name**, **Top 5 similar**, **Detection names**, and **User** columns for the entered words. The table is updated automatically to show matching entries.

Searching analysis reports

You can search data in analysis reports, such as similar samples, matched genotypes, and strings. From analysis report pages, you can also go to genotype and string properties to view and search samples that contain the genotype or string.

Searching data in analysis reports

To search data in an analysis report:

1. In the main menu, click **Analysis** or **History**.
2. In the **MD5** column, click any underlined sample hash.
3. On the analysis report page, open the **Attribution entity samples** or **Previously analyzed samples** tab depending on a sample type you want to search data for.
4. Enter one or more words you want to search in the **Search** field above any result table.

   Depending on a table you search in, Kaspersky Threat Attribution Engine searches the following columns for the entered words:

   - For the **Similar samples** table in the **Attribution entity samples** tab, search is performed in the **MD5**, **Attribution entity**, and **Aliases** columns.
   - For the **Similar samples** table in the **Previously analyzed samples** tab, search is performed in the **MD5**, **File name**, **Top 5 similar**, and **User** columns.
   - For the **Matched genotypes** table in the **Attribution entity samples** tab, search is performed in the **Genotype** and **Used by** columns.

- For the **Matched genotypes** table in the **Previously analyzed samples** tab, search is performed in the **Genotype**, **Used by**, and **File names** columns.
- For the **Matched strings** table in the **Attribution entity samples** tab, search is performed in the **String** and **Used by** columns.
- For the **Matched strings** table in the **Previously analyzed samples** tab, search os performed in **String**, **Used by**, and **File names** columns.

The table is updated automatically to show matching entries.

Searching genotype and string properties

To search properties of a genotype or a string for a specific sample:

1. In the main menu, click **Analysis** or **History**.
2. In the **MD5** column, click any underlined sample hash.
3. On the analysis report page, open the **Attribution entity samples** or **Previously analyzed samples** tab depending on a sample type you want to search data for.
4. If you want to search genotype properties, click a genotype identifier in the **Matched genotypes** table.
5. If you want to search string properties, click a string identifier in the **Matched strings** table.
6. On the property page, enter one or more words in the **Search** field above the table.

   Kaspersky Threat Attribution Engine searches for the entered words in the **MD5** and **Attribution entity** columns in the **Attribution entity samples** tab or **MD5** and **Top 5 similar** columns in the **Previously analyzed samples** tab. The table is updated automatically to show matching entries.

Searching attribution entities

You can search attribution entities, as well as samples that are attached to the attribution entities, by specifying search keywords.

Searching attribution entities

To search attribution entities:

1. In the main menu, click **Attribution entities**, and then select one of the following tabs depending on the attribution entity type you want to search:
   - **Kaspersky**—Attribution entities provided by Kaspersky
   - **Custom**—Custom attribution entities
2. Enter one or more words you want to search in the **Search** field above the table.

Kaspersky Threat Attribution Engine searches the **Name**, **Parent**, **Description**, **Aliases**, and **Comments** columns for the entered words. The table is updated automatically to show matching entries.

Searching attribution entity samples

To search samples that are attached to a specific attribution entity:

1. In the main menu, click **Attribution entities**, and then select one of the following tabs depending on the attribution entity type you want to search:

   - **Kaspersky**—Attribution entities provided by Kaspersky
   - **Custom**—Custom attribution entities

2. Click an attribution entity name.
3. In the attribution entity properties, open the **Samples** or **Attachment** tab.
4. Enter one or more words you want to search in the **Search** field above the table in the tab.

   Depending on the tab, Kaspersky Threat Attribution Engine searches the following columns for the entered words:

- In the **Samples** tab: **MD5** and **Parent** columns
- In the **Attachment** tab: **MD5**, **Parent**, **File name**, and **Owned by** columns

The table is updated automatically to show matching entries.

Exporting data

This section explains how to export Kaspersky Threat Attribution Engine analysis results and what information is contained in the exported reports.

You can export analysis results as files in TXT, JSON, and STIX formats or generate a file with YARA rules. The files contain detailed information about results of the analyses performed.

About result and unpacking codes

The exported report files contain codes that correspond to analysis and unpacking statuses displayed in the interface. For more information about the statuses, see About analysis statuses.

Result codes

The table below describes result codes that exported reports include.

Result codes

Result code	Result status
100	Similarity found
101	Canceled
4	Sample is trusted
5	File not found
6	Empty extraction
8	No similarities found
10	Extraction has failed
15	Size is not valid
16	Duplicate
50	Internal error
55	Timeout exceeded

Unpacking operation codes

The table below describes unpacking operation codes that exported reports include.

Unpacking operation codes

Unpacking operation code	Unpacking status
0	Successful
1	Unpacking operation has failed
2	Not all unpacked samples were processed because the unpacked file limit was reached
3	Sample is protected with an unknown password
4	Sample is corrupted

Generating JSON report file

To export a report to a JSON file:

1. In the main menu, click **Analysis** or **History**, and then select one or more samples using the check boxes in the first column.

   If necessary, you can navigate to analysis report, genotype properties, or string properties to export results only for one opened sample.

2. Click the **Export** button above the table and select the **JSON** option in the drop-down list that opens.

3. Save the generated JSON file using the browser window.

JSON file structure

The file includes an array that shows information about an analyzed sample and other samples related to it. For each sample, the following parameters are shown:

- **md5**—MD5 hash of the analyzed sample.
- **filename**—Name of the analyzed sample.
- **size**—Sample size, in bytes.
- **total_bad_genotypes**—Total number of genotypes in the analyzed sample that matched genotypes in similar attribution entity samples if the similarity threshold was ignored.
- **matched_bad_genotypes**—Number of genotypes in the analyzed sample that matched genotypes in similar attribution entity samples.
- **total_bad_strings**—Total number of strings in the analyzed sample that matched strings in similar attribution entity samples if the similarity threshold was ignored.
- **matched_bad_strings**—Number of strings in the analyzed sample that matched strings in similar attribution entity samples.
- **top_5**—Array of top five attribution entities that have samples with the highest similarity to the analyzed sample.

  For each attribution entity, the following parameters are shown:

  - **attribution_entity**—Attribution entity name.
  - **similarity**—Similarity percentage.
- **previously_matched_samples_count**—Number of similar samples among previously analyzed samples.
- **detection_names**—Names of detections found in the analyzed sample by the anti-virus engine.
- **analysis_date**—Date and time when the sample was analyzed, in UTC format.
- **username**—Name of the user that sent the sample for analysis.

- **result_code**—Analysis result code.

  For code descriptions, see About result and unpacking codes.

- **unpacking_status**—Unpacking operation code.

  For code descriptions, see About result and unpacking codes.

- **parent_md5**—MD5 hash of the parent file for the analyzed file.

  This parameter is used for samples unpacked from submitted archives.

- **extracted_path**—Path to the analyzed sample relative to the parent sample that was sent for analysis.

  This parameter is used for samples unpacked from submitted archives.

- **similar_attribution_entities_samples**—Array of attribution entity samples that are similar to the analyzed sample.

  For each sample, the following parameters are shown:

  - **md5**—MD5 hash of a sample.
  - **size**—Sample size, in bytes.
  - **total_genotypes**—Total number of genotypes in the similar attribution entity sample.
  - **matched_genotypes**—Number of genotypes that matched between the similar attribution entity sample and the analyzed sample.
  - **total_strings**—Total number of strings in the similar attribution entity sample.
  - **matched_strings**—Number of strings that matched between the similar attribution entity sample and the analyzed sample.
  - **similarity**—Similarity percentage between the analyzed sample and the similar attribution entity sample.
  - **attribution_entity**—Name of the attribution entity that owns the similar sample.
  - **aliases**—Aliases of the attribution entity that owns the sample.

- **matched_genotypes_with_attribution_entity_samples**—Array of genotypes in the analyzed sample that matched genotypes in similar attribution entity samples.

  For each genotype, the following parameters are shown:

  - **genotype**—Genotype in the analyzed sample that matched in similar attribution entity samples.
  - **matched**—Number of attribution entity samples that have the matched genotype.
  - **used_by**—Array of attribution entities that own samples with this genotype.

    For each attribution entity, the following parameters are shown:

    - **attribution_entity**—Attribution entity name.
    - **matched**—Number of samples owned by this attribution entity that have this genotype.

- **matched_strings_with_attribution_entity_samples**—Array of strings in the analyzed sample that matched strings in similar attribution entity samples.

- **string**—String in the analyzed sample that matched in similar attribution entity samples.
- **matched**—Number of attribution entity samples that have the matched string.
- **used_by**—Array of attribution entities that own samples with this string.

  For each attribution entity, the following parameters are shown:

  - **attribution_entity**—Attribution entity name.
  - **matched**—Number of samples owned by this attribution entity that have this string.

- **similar_previously_analyzed_samples**—Array of similar previously analyzed samples that are similar to the analyzed sample.

  For each sample, the following parameters are shown:

  - **md5**—MD5 hash of a similar previously analyzed sample.
  - **size**—Size of a similar previously analyzed sample, in bytes.
  - **filename**—Name of a similar previously analyzed sample.
  - **total_genotypes**—Total number of genotypes in the similar previously analyzed sample.
  - **matched_genotypes**—Number of genotypes that matched between the similar previously analyzed sample and the analyzed sample.
  - **total_strings**—Total number of strings in the similar previously analyzed sample.
  - **matched_strings**—Number of strings that matched between the similar previously analyzed sample and the analyzed sample.
  - **similarity**—Similarity percentage between the analyzed sample and the similar previously analyzed sample.
  - **last_analysis_date**—Date and time when the similar previously analyzed sample was last analyzed, in UTC format.
  - **username**—Name of the user that sent the sample for analysis.
  - **top_5**—Array of top five attribution entities that have samples with the highest similarity to the similar previously analyzed sample.

    For each attribution entity, the following parameters are shown:

    - **attribution_entity**—Attribution entity name.
    - **similarity**—Similarity percentage.

- **matched_genotypes_with_previously_analyzed_samples**—Array of genotypes in the analyzed sample that matched genotypes in similar previously analyzed samples.

  - **genotype**—Genotype in the analyzed sample that matched the genotype in similar previously analyzed samples.
  - **matched**—Number of previously analyzed samples that have the matched genotype.
  - **used_by**—Array of attribution entities that own samples with this genotype.

For each attribution entity, the following parameters are shown:

- **attribution_entity**—Attribution entity name.
- **matched**—Number of samples owned by this attribution entity that have this genotype.

- **filenames**—Names of the previously analyzed samples that contain this genotype.

- **matched_strings_with_previously_analyzed_samples**—Array of strings in the analyzed sample that matched strings in similar previously analyzed samples.
  - **string**—String in the analyzed sample that matched the string in similar previously analyzed samples.
  - **matched**—Number of previously analyzed samples that have the matched string.
  - **used_by**—Array of attribution entities that own samples with this string.

    For each attribution entity, the following parameters are shown:

    - **attribution_entity**—Attribution entity name.
    - **matched**—Number of samples owned by this attribution entity that have this string.
  - **filenames**—Names of the previously analyzed samples that contain this string.

Generating TXT report file

To export a report to a TXT file:

1. In the main menu, click **Analysis** or **History**, and then select one or more samples using the check boxes in the first column.

   If necessary, you can navigate to analysis report, genotype properties, or string properties to export results only for one opened sample.

2. Click the **Export** button above the table and select the **TXT** option in the drop-down list that opens.

3. Save the generated TXT file using the browser window.

TXT file structure

Each line in the file represents a set of headers separated by semicolons or a set of values for these headers separated by semicolons.

For information about the parameters shown in the report, see the descriptions of corresponding fields in Exporting analysis results to JSON.

The analysis result is shown as follows:

- Top-level headers for an analyzed sample (one line)
- Values for top-level headers that describe the analyzed sample (one line)
- Headers for similar attribution entity samples (one line, indented)

- Values that describe similar attribution entity samples (one or more lines, indented)
- Headers for matched genotypes (one line, indented)
- Values that describe matched genotypes (one or more lines, indented)
- Headers for matched strings (one line, indented)
- Values that describe matched strings (one or more lines, indented)
- Headers for similar previously analyzed samples (one line, indented)
- Values that describe similar previously analyzed samples (one or more lines, indented)
- Headers for matched genotypes with previously analyzed samples (one line, indented)
- Values that describe matched genotypes with previously analyzed samples (one or more lines, indented)
- Headers for matched strings with previously analyzed samples (one line, indented)
- Values that describe strings matched with previously analyzed samples (one or more lines, indented)

Generating STIX 2.1 report file

To export a report to a file in STIX 2.1 format:

1. In the main menu, click **Analysis** or **History**, and then select one or more samples using the check boxes in the first column.

   If necessary, you can navigate to analysis report, genotype properties, or string properties to export results only for one opened sample.

2. Click the **Export** button above the table and select the **STIX** option in the drop-down list that opens.

3. Save the generated file in the STIX 2.1 format using the browser window.

STIX file structure

For documentation about the STIX 2.1 format, see http://docs.oasis-open.org/cti/stix/v2.1/stix-v2.1.html.

Examine the following template of a STIX file:

Example:

```
{
    "type": "bundle",
    "id": "bundle--{BUNDLE_ID}",
    "objects": [
        {
            "type": "indicator",
            "spec_version": "2.1",
            "id": "indicator--{INDICATOR_ID_USER_SAMPLE}",
            "created": "{TIMESTAMP}",
            "modified": "{TIMESTAMP}",
            "name": "{NAME_USER_SAMPLE}",
            "description": "{DESCRIPTION_USER_SAMPLE}",
            "pattern": "[ file:hashes.MD5 = '{HASH}' ]",
            "pattern_type": "stix",
            "valid_from": "{TIMESTAMP}"
        },
        {
            "type": "indicator",
            "spec_version": "2.1",
            "id": "indicator--{INDICATOR_ID_ATTRIBUTION_ENTITY_SAMPLE}",
            "created": "{TIMESTAMP}",
            "modified": "{TIMESTAMP}",
            "name": "{NAME_ATTRIBUTION_ENTITY_SAMPLE}",
            "description": "{DESCRIPTION_ATTRIBUTION_ENTITY_SAMPLE}",
            "pattern": "[ file:hashes.MD5 = '{HASH}' ]",
            "pattern_type": "stix",
            "valid_from": "{TIMESTAMP}"
        },
        {
            ...
        }
    ]
}
```

The following fields of a STIX file template are possible:

- **type**—Type of the described object in a STIX model.

  The objects in the generated STIX file have type `"type": "indicator"`.

- **id**—Unique identifier of the described object in the STIX model.

- **objects**—STIX objects array in the format `"objects": [ ... ]`.

- **spec_version**—STIX specification used in the file. Only version 2.1 is supported.

- **created**—Date and time when the object was created. For example, `"created": "2020-04-06T20:03:48.000Z"`.

- **modified**—Date and time when the object was last modified. For example, `"modified": "2020-04-06T22:53:49.213Z"`.

- **pattern**—Pattern used to describe the objects with the type *indicator* (indicator of comprometation). For example, `"pattern": "[ file:hashes.MD5 = '01cc483dcec000004ec91f0413714' ]"`.

- **pattern_type**—Pattern language used to describe objects with the type *indicator*.

  The objects in the generated STIX file have pattern type `"pattern_type": "stix"`.

- **name**—Name of an object with the type *indicator*. For example, `"name": "01cc483dcec6e09f8504ec91f0413714"`.

- **description**—Any additional information about the object. For example:

  ```
  "description":
  "file_name=01cc483dce000000504ec91f0413714;top_5_similar=MyXXXloiter(99%);01cc483dcec6e
  is similar to files: 508cd619ce9ede44be8XXXX0a1ba",
  ```

```
"description": "file_name=17AEC1E7300000040941F13D503;
top_5_similar=Sh__a(46%),G__p(37%),Phant__nce(28%);17aec1e000b0a140941f13d503 is
similar to files: eeecfa2900000deb8029d27db125e, 29779de954e5c000008f3b720dd7e8c5b,
5eebfd977cffc0a7b2732f2aa3e1cb0b, 10267c8597000000eb4e5a91a,
5790bf00000003eacecf319cf2, 7d73e8b95180000002c5e59f26021,
64f33a000000d4ef5a7c32eef96",
"description": "attribution_entity=MyExploiter",
"attribution_entity=GazaTeam; aliases=DownExecute,Gaza hacker team"
```

- **valid_from**—Date and time from which the object with a type *indicator* becomes a valid indicator of the identified object. For example, `"valid_from": "2021-01-01T00:00:00.001Z"`.

Generating YARA rules

You can generate YARA rules from samples sent for analysis. Kaspersky Threat Attribution Engine uses strings and genotypes extracted during initial file processing as a source data to generate YARA rules. As Kaspersky Threat Attribution Engine does not use similarity data and attribution entity database, YARA rules generation does not depend on the similarity check procedure.

To generate a YARA rule file:

1. In the main menu, click **Analysis** or **History**, and then select one or more samples using the check boxes in the first column.

   If necessary, you can navigate to analysis report, genotype properties, or string properties to export results only for one opened sample.

2. Click the **Export** button above the table and select the **YARA** option in the drop-down list that opens.

3. In the **YARA rules** window that opens, specify the following parameters:

   - **Name**—Name of the file with YARA rules.
   - **Description**—Description of the generated YARA rule file.
   - Select one of the following YARA rule type to generate:
     - **Simple rules**—Generate simple rules only, which are not resource-intensive, but have minor resumptive value. If you select several files, simple rules are generated from *individual* genotypes or strings extracted from each sample in the group.
     - **Cluster rules**—Generate cluster rules only, which are resource-intensive and might result unexpectedly due to limited computing capability of the server. Cluster rules cover more than one file and have a major resumptive value compared to the simple rules. Cluster rules are generated from *matched* genotypes or strings extracted from the selected samples in the group.

       > You must select at least two samples to generate cluster rules. Maximum number of samples to generate cluster YARA rules is 100. If you select more than 100 samples, the corresponding error message is displayed.

     - **All rules**—Generate both simple rules and cluster rules.

If you export data from analysis report, genotype properties, or string properties, you can generate only simple rules.

- **Condition by genotypes**—Minimum number of genotypes that must match between samples in a group to generate the cluster YARA rule.
- **Condition by strings**—Minimum number of strings that must match between samples in a group to generate the cluster YARA rule.

4. Click **Ok** to generate YARA rules.

If Kaspersky Threat Attribution Engine could not generate YARA rules, a message *Selected samples do not contain strings or genotypes for generating YARA rules* is displayed at the lower part of the page. This can happen due to one of the following reasons:

- You selected to generate only cluster rules, but selected samples do not have enough matching genotypes and strings.
- All genotypes and strings from the selected samples are considered trusted and popular.
- None of the selected samples contain genotypes and strings to generate YARA rules.

If Kaspersky Threat Attribution Engine could not generate simple YARA rules from any of the selected samples because the samples do not contain genotypes or strings to generate the rules, a message *<N> errors occurred while generating YARA rules. Show errors* is displayed in the lower part of the page, where <N> is the number of errors. You can click the **Show errors** link to open a pane containing the following information:

- List of samples from which YARA rules could not be generated
- Description of the error, which explains why simple YARA rules were not generated

YARA rule structure

For YARA rule specification, see https://yara.readthedocs.io/en/stable/writingrules.html.

Generated YARA rules contain the following fields:

- **meta**—Metadata of the generated YARA rule.

  The metadata includes the following fields:

  - **description**—Description you specified when generating YARA rules.
  - **last_modified**—Date and time when the rules were generated, in UTC format. For example, 2020-05-17 13:31:29.
  - **author**—User account used to generate the rule.
  - **hash**—MD5 hash of the sample from which genotypes and strings were extracted.
  - **version**—YARA rule version. Only version 1.0 is supported.

- **strings**—YARA genotypes and strings that are related to the sample or samples for the cluster rules.

  String example:

```
$a0 = "__Z8proc_cmdPcPhPj" fullword ascii
$a1 = "hs->count < 64" fullword wide
```

Genotype example:

```
$c0 = {547261636b6572282255412d39343536}
$c1 = {53717200576e00006400005328090000}
```

- **condition**—A set of the following conditions combined using `and`:

  - **uintM(N) == 0xXYZ**, where `M` is the number of bits, `N` is shifting, `XYZ` is HEX (little endian) value from the list of known signatures in the upper case.

    This is an optional condition for checking the file header. The header is checked if the sample signature matched the value from the list of known signatures. Otherwise, the condition is not used.

    For example, `uint16(0) == 0x5A4D`.

  - **filesize < <file size>**, where <file size> is the size of the file used to extract genotypes and strings, multiplied by 4, and rounded up to a `d*` type number. `d` is any number from 1 to 9 and `*` is any number of nulls.

    For example, `filesize < 500000`.

  - **N of them**, where <N> is the minimum number of genotypes or strings in the rule as set in the **Condition by genotypes** and **Condition by strings** parameters.

    For example, `2 of them`.

Viewing Kaspersky Threat Attribution Engine settings

Depending on your user role, you have access to a different set of settings tab and functionality available in the **Settings** section:

- Regular users can only view Kaspersky Threat Attribution Engine information in the **About** tab and license information in the **License** tab.
- Users with administrator rights can view Kaspersky Threat Attribution Engine information in the **About** tab and license information in the **License** tab, as well as configure Kaspersky Threat Attribution Engine and resolve update conflicts. For more information, see Configuring Kaspersky Threat Attribution Engine settings and Resolving update conflicts.

Viewing the current license information

To view information about your current Kaspersky Threat Attribution Engine license,

In the main menu, click **Settings**, and then open the **License** tab.

This tab displays information about the current license that is active on your account. If the license is expired, the corresponding notification is displayed. For more information about the tab content, see Viewing your current license.

Viewing Kaspersky Threat Attribution Engine information

To view information about Kaspersky Threat Attribution Engine,

In the main menu, click **Settings**, and then open the **About** tab.

The table below describes information that is shown in this tab.

About tab

Field	Description
KTAE version	Current version of Kaspersky Threat Attribution Engine
KTAE database version	Current version of Kaspersky Threat Attribution Engine databases
KTAE database last update	Date and time of the last Kaspersky Threat Attribution Engine database update, in Coordinated Universal Time (UTC)
AV database last update	Date and time of the last Kaspersky Threat Attribution Engine anti-virus database updates, in Coordinated Universal Time (UTC)

Administering Kaspersky Threat Attribution Engine

This section explains how to administer Kaspersky Threat Attribution Engine.

Only administrators have access to this functionality.

Launching admin_cli utility

This section describes the admin_cli utility and how to use it.

The admin_cli is a utility for administrators that they can use to manage Kaspersky Threat Attribution Engine. With the utility, administrators can do the following:

- Manage the Kaspersky Threat Attribution Engine service
- Manage user accounts
- Update Kaspersky Threat Attribution Engine components
- Manage API keys

Launching admin_cli

To launch the admin_cli utility:

1. Launch the admin_cli utility as follows:

   ```
   sudo bash /opt/ktae/bin/admin_cli
   ```

   The utility connects to the database, and the commands in the table below become available.

2. Execute a command to manage with Kaspersky Threat Attribution Engine.

   The commands available in the admin_cli utility are described in the section below.

admin_cli commands

The table below describes the commands available in the admin_cli utility.

admin_cli commands

Full command	Short command	Description
list	l	Display a list of Kaspersky Threat Attribution Engine users and information about each account: user account identifier, user name, and user role.
create	c	Create a user account.
delete	d	Delete a user account.
grant-role	gr	Assigns a role to a user account: `admin` or `user`.
revoke-role	rr	Revoke a role from a user account.
change-password	cp	Change the password of a user.

add-key	ak	Add an API key to a user account.
delete-key	dk	Delete all API keys from a user account.
check-key	ck	Check if the API key is valid for a user account.
download-all-updates	dau	Download updates for a Kaspersky Threat Attribution Engine database, binaries and anti-virus databases. This command is available only if you installed Kaspersky Threat Attribution Engine with a simple configuration.
aggregate-update	u	Update the Kaspersky Threat Attribution Engine database, binaries, and anti-virus databases. Before running this command, you should download update files for the components.
update-data	ud	Update only the Kaspersky Threat Attribution Engine database. Before running this command, you should download update files for the components.
activate-scanner-bases	asb	Update the anti-virus databases. Before running this command, you should download update files for the components.
update-scanner-bases	usb	Download updates for the anti-virus databases and apply them. This command is available only if you installed Kaspersky Threat Attribution Engine with a simple configuration.
run-service	rs	Launch the Kaspersky Threat Attribution Engine service.
terminate-service	ts	Stop the Kaspersky Threat Attribution Engine service.
quit	q	Exit the admin_cli utility.
help	h	Display a help message with a command list.

Managing the Kaspersky Threat Attribution Engine service

This section explains how to start and stop the Kaspersky Threat Attribution Engine service using the admin_cli utility.

Starting the Kaspersky Threat Attribution Engine service

To start the Kaspersky Threat Attribution Engine service:

1. Launch the admin_cli utility.
2. Run the following command to start the Kaspersky Threat Attribution Engine service:

   ```
   run-service
   ```

Stopping the Kaspersky Threat Attribution Engine service

To stop the Kaspersky Threat Attribution Engine service:

1. Launch the admin_cli utility.
2. Run the following command to stop the Kaspersky Threat Attribution Engine service:

   ```
   terminate-service
   ```

Configuring Kaspersky Threat Attribution Engine settings

This section explains how to configure Kaspersky Threat Attribution Engine parameters and settings.

To configure Kaspersky Threat Attribution Engine settings,

1. In the main menu, click **Settings**, and then open the **System settings** tab.
2. If necessary, specify the following parameters in the **YARA rule settings** group to configure conditions for generating YARA rules:
   - **Condition by genotypes**—Specify the minimum number of genotypes that must match between samples in a group to generate a cluster YARA rule.

     By default, the parameter is set to 10.
   - **Condition by strings**—Specify the minimum number of strings that must match between samples in a group to generate a cluster YARA rule.

     By default, the parameter is set to 10.
3. If necessary, specify the following parameters in the **Previously analyzed samples** group to configure the conditions to be used when searching for a similar previously analyzed sample:
   - **Condition by genotypes**—Specify the genotype threshold for all previously analyzed samples. An analyzed sample and a previously analyzed samples are considered similar if they have more matching genotypes than specified in this parameter.

     By default, the parameter is set to 2.
   - **Condition by strings**—Specify the string threshold for all previously analyzed samples. An analyzed sample and a previously analyzed samples are considered similar if they have more matching strings than specified in this parameter.

     By default, the parameter is set to 2.
4. Click **Save**.

Currently, the **Storage period (days)** parameter in the **History settings** group cannot be changed. The parameter is set to 180 days.

Managing user accounts

This section describes how to manage Kaspersky Threat Attribution Engine user accounts using the admin_cli utility.

Viewing a list of user accounts

To view a list of user accounts:

1. Launch the admin_cli utility.
2. Run the following command:

   ```
   list
   ```

The utility displays a list of user accounts with user account identifier, user name, and user role for each account.

Creating a user account

To create a user account:

1. Launch the admin_cli utility.
2. Run the following command to enter the dialog mode:

   `create`
3. In the dialog mode, enter the following information to the entry fields:
   - In the `Login` field, enter login for the user account.
   - In the `Username` field, enter the user account name, which will be displayed in the web interface.
   - In the `Password` field, enter the password for the user account.
   - In the `Role` field, enter role for the user account: `user` for a regular user or `admin` for a user with administrator rights.

   The user account with the specified parameters is created.

Deleting a user account

To delete a user account:

1. Launch the admin_cli utility.
2. Run the following command to enter the dialog mode:

   `delete`
3. In the `User id` field, enter the identifier of the user account to delete.

   The user account with the specified identifier is deleted.

Changing a user account password

To change a user account password:

1. Launch the admin_cli utility.
2. Run the following command to enter the dialog mode:

   `change-password`
3. In the dialog mode, enter the following information to the entry fields:
   - In the `User id` field, enter the identifier of the user account for which you want to update password.
   - In the `New password` field, enter a new password for the account.

The password for the user account is updated.

Assigning a role to a user account

To assign a new role to a user account:

1. Launch the admin_cli utility.
2. Run the following command to enter the dialog mode:

   `grant-role`
3. In the dialog mode, enter the following information to the entry fields:
   - In the `User id` field, enter the identifier of the user account for which you want to change the user role.
   - In the `New role` field, enter a new role for the account: `user` for a regular user or `admin` for a user with administrator rights.

   The role for the user account is updated.

Revoking a role from a user account

To revoke a role from a user account:

1. Launch the admin_cli utility.
2. Run the following command to enter the dialog mode:

   `revoke-role`
3. In the dialog mode, enter the following information to the entry fields:
   - In the `User id` field, enter the identifier of the user account for which you want to revoke the user role.
   - In the `Role to revoke` field, enter a role to revoke: `user` for a regular user or `admin` for a user with administrator rights.

   The specified role is revoked from the user account.

Updating Kaspersky Threat Attribution Engine components

This section explains how to update Kaspersky Threat Attribution Engine components, which include Kaspersky Threat Attribution Engine databases, binary files, and anti-virus databases.

> We recommend that you update Kaspersky Threat Attribution Engine databases, binary files, and anti-virus databases every 14 days. You can check the date of the last update in the **Settings** section.

Updating Kaspersky Threat Attribution Engine includes the following stages:

1. Backing up the database.

2. Downloading updates for the components.

3. Updating Kaspersky Threat Attribution Engine components.

If you used the simple configuration to install Kaspersky Threat Attribution Engine, you can enable the anti-virus databases to update automatically.

Backing up the database

You need to back up Kaspersky Threat Attribution Engine databases before updating to prevent loss of personal data.

To back up the databases, do one of the following:

- For tables over 100 megabytes (MB) in size, use the `FLUSH TABLES <table name> ... FOR EXPORT` command.

  Replace `<table name>` with the name of a table to back up. For more information about this command, refer to the MySQL 8.0 Reference Manual.

- For tables under 100 MB in size, use the mysqldump client utility.

  For more information about this utility, refer to the MySQL 8.0 Reference Manual.

We recommend that you back up the following tables:

- analyzes
- analyzing_sample_genotype_relations
- analyzing_sample_genotypes
- analyzing_sample_meta_data
- analyzing_sample_similar_samples
- analyzing_sample_statistic
- analyzing_sample_string_relations
- analyzing_sample_strings
- analyzing_samples
- api_keys
- attaching_samples
- attribution_entities
- conflicts
- credentials
- current_scanning_bases

- customer_attribution_entity_sample_genotypes
- customer_attribution_entity_sample_strings
- customer_attribution_entity_samples
- flyway_schema_history
- locked_files
- previously_analyzed_sample_genotypes
- previously_analyzed_sample_strings
- previously_analyzed_sample_samples
- scanning_bases
- sessions
- system_settings
- update_metadata
- updates_states
- users

Downloading updates

Depending on the configuration you used to install Kaspersky Threat Attribution Engine, the updates for its components should be downloaded differently.

To download updates for Kaspersky Threat Attribution Engine with the simple configuration:

1. In the open perimeter, navigate to the following directory:

   ```
   cd /opt/ktae-updater/
   ```

2. Run the following script to download the updates:

   ```
   sudo ./download_all.sh <OPEN_UPDATES_DIR>
   ```

   Replace `<OPEN_UPDATES_DIR>` with a path to the directory where the downloaded files will be stored. This directory was created when preparing to install Kaspersky Threat Attribution Engine.

   If the Kaspersky Threat Attribution Engine license key is not valid, the following error occurs: `Can't initialize engine. License problem. Failed to download Scanner updates! Result code: 93.`
   Make sure the license key for your account is valid.

To download updates for Kaspersky Threat Attribution Engine with the air-gapped configuration:

1. In the open perimeter, navigate to the following directory:

```
cd /opt/ktae-updater/
```

2. Run the following script to download the updates:

   ```
   sudo ./download_all.sh <OPEN_UPDATES_DIR>
   ```

   Replace `<OPEN_UPDATES_DIR>` with a path to the directory where the downloaded files will be stored. This directory was created when preparing to install Kaspersky Threat Attribution Engine.

   If the Kaspersky Threat Attribution Engine license key is not valid, the following error occurs: `Can't initialize engine. License problem. Failed to download Scanner updates! Result code: 93.`
   Make sure the license key for your account is valid.

3. Copy the downloaded files to the `<SECURE_UPDATES_DIR>` directory created in the secure perimeter when preparing to install Kaspersky Threat Attribution Engine.

   Make sure to keep the directory structure as it is in the `<OPEN_UPDATES_DIR>`.

Updating the components

Depending on the configuration you used to install Kaspersky Threat Attribution Engine, run the steps in different perimeters:

- For the simple configuration, run the steps in the open perimeter.
- For the air-gapped configuration, run the steps in the separate secure perimeter.

To update Kaspersky Threat Attribution Engine components:

1. Launch the admin_cli utility.
2. Run the following command to stop the Kaspersky Threat Attribution Engine service:

   ```
   terminate-service
   ```

3. Run one of the following commands to update one or more of the Kaspersky Threat Attribution Engine components:

   - To update all components:

     ```
     aggregate-update
     ```

   - To update only Kaspersky Threat Attribution Engine databases:

     ```
     update-data
     ```

   - To update only anti-virus databases:

     ```
     activate-scanner-bases
     ```

     You can also enable the anti-virus databases to update automatically.

   The components are updated with the update files that you downloaded using the instructions in Downloading updates.

4. Run the following command to start the Kaspersky Threat Attribution Engine service:

   ```
   run-service
   ```

If update conflicts happen during the update, you can resolve them in the **Settings** section.

Enabling automatic updates for anti-virus databases

If you installed Kaspersky Threat Attribution Engine using the simple configuration, you can enable anti-virus databases to be updated automatically. With automatic updates enabled, Kaspersky Threat Attribution Engine checks for updates in anti-virus database every four hours and updates them, if necessary. By default, automatic updates for the anti-virus databases are disabled.

To enable automatic update for the anti-virus databases:

1. Run the following command to stop the Kaspersky Threat Attribution Engine service:

   ```
   terminate-service
   ```

2. Open the /opt/ktae/conf/application.conf configuration file.

3. In the `service-settings` > `scanning-settings` > `scanning-bases-settings` > `update` section, set the `auto` parameter to `true` as follows

   ```
   service-settings {
     ...
     scanning-settings {
       scanning-bases-settings {
         update {
           auto = true // <-- change this value to "true"
         }
       }
     }
   }
   ```

4. Run the following command to start the Kaspersky Threat Attribution Engine service:

   ```
   run-service
   ```

Resolving update conflicts

This section explains what update conflicts are and how to resolve them.

About update conflicts

An update conflict happens when trusted samples provided with the latest Kaspersky Threat Attribution Engine update match existing samples that are attached to an attribution entity. In this case, Kaspersky Threat Attribution Engine cannot resolve such conflict automatically. Users with administrator rights must manually determine whether the sample is considered trusted or not.

> Update conflicts do not block the update operation. Kaspersky Threat Attribution Engine can update databases even if it detects an update conflict.

If Kaspersky Threat Attribution Engine has active update conflicts, a corresponding notification is shown in the notification pane. Unread notifications are indicated with a notification icon  in the main menu. You can click the icon to see the notification message and go to the **Update conflicts** tab in the **Settings** section to resolve the update conflict.

Only users with administrator rights receive the notification about update conflicts.

Resolving update conflicts

To resolve update conflicts:

1. Log in to Kaspersky Threat Attribution Engine as a user with administrator rights.
2. In the main menu, click **Settings**, and then open the **Update conflicts** tab.

   The tab displays the list of update conflicts and the update dates.

   You can also click the notification icon in the main menu and click **Update conflicts** link in the opened notification about the conflict.
3. Select update conflicts you want to resolve with check boxes.
4. Click **Resolve**.
5. In the **Resolve conflict** window, select how you want to resolve the selected conflicts:

   - **Move selected samples to trusted samples**—Detach the selected samples from their current attribution entities and move them to trusted samples.
   - **Keep selected samples as attribution entity samples**—Keep the selected samples attached to their current attribution entities. The samples will not be considered trusted by Kaspersky Threat Attribution Engine.
6. Click **Resolve**.

Kaspersky Threat Attribution Engine REST API Reference

This section describes the REST API of Kaspersky Threat Attribution Engine.

Authentication in Kaspersky Threat Attribution Engine API

This section explains how to authenticate to Kaspersky Threat Attribution Engine REST API using API keys.

Managing API keys

Kaspersky Threat Attribution Engine REST API requires API keys for authentication. You must get an API key for a user account to be able to make requests using API. Use the admin_cli utility to manage API keys for user accounts.

> Only administrators can add API keys to user accounts. We recommend that you create separate accounts for each user to interact with Kaspersky Threat Attribution Engine API only. Working with Kaspersky Threat Attribution Engine API using the same account you use for regular interaction is not recommended.

Creating an API key

To create an API key:

1. Launch the admin_cli utility.
2. Run the following command to enter the dialog mode:

   `add-key`
3. In the `User id` field, enter the identifier of a user account to which you want to add an API key.

   If the API key for this user is generated successfully, a message appears that describes the command result.
4. Save the generated API key to share it with the user.

Checking API key validity

To check the validity of an API key:

1. Launch the admin_cli utility.
2. Run the following command to enter the dialog mode:

   `check-key`
3. In the dialog mode, enter the following information to the entry fields:

   - In the `User id` field, enter the identifier of a user account for which you want to check API key.

- In the `Api-Key value` field, enter the user API key.

If the entered key is valid, a message `Valid Api-Key` appears. If the key is invalid, a message `Api-key is invalid` appears.

Deleting API keys

To delete all API keys for a user account:

1. Launch the admin_cli utility.
2. Run the following command to enter the dialog mode:

   ```
   delete-key
   ```

3. In the `User id` field, enter the identifier of a user account to which you want to delete API keys.

   If the API key for this user is deleted successfully, a message appears that describes the command result.

Authenticating with API key

To authenticate in Kaspersky Threat Attribution Engine REST API, you must have an API key generated for your account. Contact your administrator to obtain your API key.

To authenticate using an API key:

1. In a request header, add an `Api-Key` header using the following format:

   ```
   Api-Key: <key>
   ```

   Replace the `<key>` with the API key that is provided from your administrator.

2. Make any API request and check the response status:

   - Status `200` means that the request was successful.
   - Status `401` means your authorization attempt was not successful.

     Make sure your API key is correct and valid. You can contact your administrator for assistance.

Example of API request

This section shows an example of sending a sample for analysis and receiving a report with possible errors processing using Kaspersky Threat Attribution Engine API. For description of methods used in the example, examine the Methods section.

```
import time
import requests
service_url = 'https://ktae-service.com/api'
api_key_header = {'api-key': '1797611e-7e3e-454d-b51b-7610272a08bb'}
def start_analysis():
    url = f'{service_url}/analysis'
    # up to 100 files
```

```
    files = [
        ('data', ('sample1', open('<path to sample1>', 'rb'), 'text/html')),
        ('data', ('sample2', open('<path to sample2>', 'rb'), 'text/html'))
    ]
    while True:
        response = requests.post(url, headers=api_key_header, files=files, verify=False,
timeout=30)
        if response.status_code == 200:
            break
        # if analysis queue limit is exceeded then sleep and retry
        if response.status_code == 503 and response.headers.get('content-type', None) ==
'application/json' \
                and response.json().get('code', None) == 545:
            time.sleep(2)
        else:
            raise RuntimeError(f"Failed to start analysis for sample(s):" +
                               f" POST '{url}' returned HTTP{response.status_code},
response body: {response.text}")
    analysis_id = response.json()['analysisId']
    return analysis_id
def get_report(analysis_id):
    url = f'{service_url}/report/{analysis_id}?format=json'
    while True:
        response = requests.get(url, headers=api_key_header, verify=False, timeout=30)
        if response.status_code == 200:
            break
        # if analysis is in progress then sleep and retry
        if response.status_code == 202 and response.headers.get('content-type', None) ==
'application/json' \
                and response.json().get('code', None) == 540:
            time.sleep(2)
        # if analysis is in queue then sleep and retry
        if response.status_code == 202 and response.headers.get('content-type', None) ==
'application/json' \
                and response.json().get('code', None) == 544:
            time.sleep(2)
        else:
            raise RuntimeError(f"Sample(s) analysis failed:" +
                               f" GET '{url}' returned HTTP{response.status_code},
response body: {response.text}")
    return response.json()
def start_analysis_and_get_report():
    return get_report(start_analysis())
if __name__ == '__main__':
    start_analysis_and_get_report()
```

Methods

This section describes methods available in Kaspersky Threat Attribution Engine REST API.

Attribution entity methods

This section describes the attribution entity methods.

POST attribution-entities

The `attribution-entities` method creates a custom attribution entity.

Path

`/api/attribution-entities`

Request method

`POST`

Request parameters

This request has no parameters.

Request body

Media type: `application/json`.

POST attribution-entities request body

Name	Required	Schema	Description
`AttributionEntity`	Yes	`AttributionEntity`	Attribution entity parameters

Responses

POST attribution-entities request responses

Code	Media type	Response schema	Description
`200`	`application/json`	`SuccessfulResponse`	OK. The attribution entity was successfully created.
`202`	`application/json`	`FailureResponse`	Accept. The request has been accepted for processing, but the processing has not been completed. The response contains the description and code of an error.
`400`	`application/json`	`MalformedInputResponse`	Bad request. The response contains the description of an error.
`401`	`application/json`	`ErrorResponse`	Unauthorized. Authorize using your API key.
`500`	`application/json`	`FailureResponse`	Server error. The response contains the description and code of an error.

Method examples

Python example

```
import requests
def create_attribution_entity():
    attribution_entities_url = 'https://ktae-service.com/api/attribution-entities'
    headers = {'api-key': '1832611e-7e3e-454d-b51b-7610232a08bb'}
    json = {
        'name': 'entity',
        'parent': 'parent',
        'description': 'ktae test entity description',
        'comments': 'ktae test entity comment',
        'minSimGenotypesThreshold': 2,
        'minSimStringsThreshold': 2,
        'reportLinks': 'https://reports.example.com/entity,https://reports.example.com/aliasentity',
        'aliases': 'aliasentity'
    }

    api_response = requests.post(url=attribution_entities_url, json=json, headers=headers)
    return api_response
```

cURL example

```
curl -X 'POST' \
  'https://ktae-service.com/api/attribution-entities' \
  -H 'accept: application/json' \
  -H 'api-key: 1832611e-7e3e-454d-b51b-7610232a08bb' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "entity",
    "parent": "parent",
    "description": "ktae test entity description",
    "comments": "ktae test entity comment",
    "minSimGenotypesThreshold": 2,
    "minSimStringsThreshold": 2,
    "reportLinks": "https://reports.example.com/entity,https://reports.example.com/aliasentity",
    "aliases": "aliasentity"
  }'
```

POST samples

The `samples` method attaches samples to a specified attribution entity. Maximum size of the sample file is 500 MB.

Path

`/api/samples/{entityName}`

Request method

`POST`

Request parameters

POST samples request parameters

Name	Required	Parameter type	Value type	Description
`entityName`	Yes	Path	String	Name of an existing attribution entity. The sample will be attached to this attribution entity.
`unpack`	No	Query	Boolean	Flag to unpack the sample. Default value: `false`.
`unpackPassword`	No	Query	String	Custom password to unpack the sample when attaching to the attribution entity. Maximum password length is 256 symbols. If you do not specify the password, Kaspersky Threat Attribution Engine will use several default passwords, such as `infected`, `virus`, `malware`, `password`, `qwerty`, to try unpacking the sample. For a complete list of default passwords, see a file located at /opt/ktae/tools/linux/scanner/passwords. This parameter is used only if the request contains the `unpack` parameter set to `true`.

Request body

Media type: `multipart/form-data`.

POST samples request body

Name	Required	Schema	Description
`File`	Yes	`File`	File contents

Responses

POST samples request responses

Code	Media type	Response schema	Description
`200`	`application/json`	`SuccessfulResponse`	OK The sample was successfully attached.
`400`	`application/json`	`ErrorResponse`	Bad request The response contains the description of an error.
`401`	`application/json`	`ErrorResponse`	Unauthorized. Authorize using your API key.
`500`	`application/json`	`FailureResponse`	Server error. The response contains the description and code of an error.

Method examples

Python example

```
import requests
def attach_sample(attribution_entity_name):
    attach_url = f'https://ktae-service.com/api/samples/{attribution_entity_name}?
unpack=true&unpackPassword=qwerty'
    files = {'data': open('<path to sample>', 'rb')}
    headers = {'api-key': '1832611e-7e3e-454d-b51b-7610232a08bb'}
    api_response = requests.post(url=attach_url, headers=headers, files=files)
    return api_response
```

cURL example

```
curl -X 'POST' \
  'https://ktae-service.com/api/samples/{attribution_entity_name}?unpack=true&unpackPassword=qwerty' \
  -H 'accept: application/json' \
  -H 'api-key: 1832611e-7e3e-454d-b51b-7610232a08bb' \
  -H 'Content-Type: multipart/form-data' \
  -F 'data=@<path to sample>'
```

Analysis methods

This section describes the analysis methods.

POST analysis

The `analysis` method starts a new analysis of a file or several files in asynchronous mode. Maximum total size of the files in a batch that are sent for analysis must not exceed 500 MB.

Path

`/api/analysis`

Request method

`POST`

Request parameters

POST analysis request parameters

Name	Required	Parameter type	Value type	Description
`resetThreshold`	No	Query	Boolean	Flag to ignore similarity thresholds for compared samples. Default value: `false`.
`unpack`	No	Query	Boolean	Flag to unpack the sample. Default value: `false`.
`amongPreviously`	No	Query	Boolean	Flag to search among previously analyzed samples Default value: `false`.
`unpackPassword`	No	Query	String	Custom password to unpack the samples sent for analysis. Maximum password length is 256 symbols. If you do not specify the password, Kaspersky Threat Attribution Engine will use several default passwords, such as `infected`, `virus`, `malware`, `password`, `qwerty`, to try unpacking the sample. For a complete list of default passwords, see a file located at /opt/ktae/tools/linux/scanner/passwords. This parameter is used only if the request contains the `unpack` parameter set to `true`.

Request body

Media type: `multipart/form-data`.

POST analysis request body

Name	Required	Schema	Description
`Files`	Yes	`Files`	File contents

Responses

POST analysis request responses

Code	Media type	Response schema	Description
200	application/json	AnalysisResponse	OK. The analysis was successfully created.
400	application/json	ErrorResponse	Bad request. The response contains description of an error.
401	application/json	ErrorResponse	Unauthorized. Authorize using your API key.
403	application/json	FailureResponse	Forbidden. Weekly analysis limit is exceeded. The response contains the description and code of an error.
429	application/json	FailureResponse	Too many requests. Analysis queue limit is exceeded. The response contains the description and code of an error.
500	application/json	FailureResponse	Server error. The response contains the description and code of an error.
503	application/json	FailureResponse	Service unavailable. Analysis limit is exceeded. The response contains the description and code of an error.

Method examples

Python example

```
import requests
def analysis():
    analysis_url = 'https://ktae-service.com/api/analysis?
resetThreshold=false&unpack=true&amongPreviously=false&unpackPassword=qwerty'
    files = {'data': open('<path to sample>', 'rb')}
    headers = {'api-key': '1832611e-7e3e-454d-b51b-7610232a08bb'}
    api_response = requests.post(url=analysis_url, headers=headers, files=files)
    return api_response
```

cURL example

```
curl -X 'POST' \
  'https://ktae-service.com/api/analysis?
resetThreshold=false&unpack=true&amongPreviously=false&unpackPassword=qwerty' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -H 'api-key: 1832611e-7e3e-454d-b51b-7610232a08bb' \
  -F 'data=@<path to sample>'
```

GET report

The `report` method exports report data in a specified format.

Path

`/api/report/{analysisId}`

Request method

`GET`

Request parameters

GET report request parameters

Name	Required	Parameter type	Value type	Description
`format`	Yes	Query	String	Report format. Possible values: `json, txt, stix.`
`analysisId`	Yes	Path	String	ID of the analysis for which you want to export results. The ID is returned by `POST analysis` method.
`Accept`	No	Header	String	Response format. Possible values: • If the parameter is specified as `Accept: application/json` and request parameter `format` is `json`, the response is in the application/json format. • If the parameter is specified as `Accept: application/octet-stream`, the response is in the application/octet-stream format. • If `Accept` is not specified, the response is in the application/octet-stream format. • If `Accept` is any other value, code 400 is returned.

Responses

GET report request responses

Code	Media type	Response schema	Description
`200`	`application/octet-`	ReportData	OK.

	stream		The report was successfully generated.
202	application/json	FailureResponse	Accept. The request has been accepted for processing, but the processing has not been completed. The response contains the description and code of an error.
400	application/json	ErrorResponse	Bad request. The response contains the description of an error.
401	application/json	ErrorResponse	Unauthorized. Authorize using your API key.
404	application/json	FailureResponse	Not found. Requested analysis was not found. The response contains the description and code of an error.
500	application/json	FailureResponse	Server error. The response contains the description and code of an error.

Method examples

Python example

```
import requests
def get_report(analysis_id):
    report_url = f'https://ktae-service.com/api/report/{analysis_id}?format=json'
    headers = {
        'api-key': '1832611e-7e3e-454d-b51b-7610232a08bb',
        'Accept': 'application/octet-stream'
    }
    api_response = requests.get(url=report_url, headers=headers)
    return api_response
```

cURL example

```
curl -X 'GET' \
  'https://ktae-service.com/api/report/{analysis_id}?format=json' \
  -H 'api-key: 1832611e-7e3e-454d-b51b-7610232a08bb'
```

POST history/report

The `history/report` method receives analysis history in a specified format.

Path

`/api/history/report`

Request method

POST

Request parameters

POST history/report request parameters

Name	Required	Parameter type	Value type	Description
format	Yes	Query	String	Report format. Possible values: json, txt, stix.

Request body

Media type: application/json.

POST history/report request body

Name	Required	Schema	Description
HistoryReportFilter	Yes	HistoryReportFilter	Report generation filter

Responses

POST history/report request responses

Code	Media type	Response schema	Description
200	application/octet-stream	ReportData	OK. The report was successfully generated.
400	application/json	ErrorResponse	Bad request. The response contains description of an error.
401	application/json	ErrorResponse	Unauthorized. Authorize using your API key.
406	application/json	ReportLimitExceed	Not acceptable. Total number of samples in the report exceeded the analysis limit.
500	application/json	FailureResponse	Server error. The response contains the description and code of an error.

Method examples

Python example

```
import requests
def get_report_history():
    report_history_url = 'https://ktae-service.com/api/history/report?format=json'
    json = {
        'startTime': 1620990460532,
        'endTime': 1620991560532,
        'userIds': [
            '92ca7dd8-a10a-11eb-983a-f48e3897cb55',
            '9da1b330-a10a-11eb-8351-f48e3897cb55'
        ],
       'minSimilarity': 5,
       'name': 'filename'
    }
    headers = {'api-key': '1832611e-7e3e-454d-b51b-7610232a08bb'}
    api_response = requests.post(url=report_history_url, headers=headers, json=json)
    return api_response
```

cURL example

```
curl -X 'POST' \
  'https://ktae-service.com/api/history/report?format=json' \
  -H 'api-key: 1832611e-7e3e-454d-b51b-7610232a08bb' \
  -H 'Content-Type: application/json' \
  -d '{
    "userIds": [
        "92ca7dd8-a10a-11eb-983a-f48e3897cb55",
        "9da1b330-a10a-11eb-8351-f48e3897cb55"
    ],
    "startTime": 1620990460532,
    "endTime": 1620991560532,
    "minSimilarity": 5,
    "name": "filename"
  }'
```

POST rules/report

The `rules/report` method generates YARA rules that are based on the input data.

Path

`/api/rules/report`

Request method

`POST`

Request parameters

This request has no parameters.

Request body

Media type: `application/json`.

POST rules/report request body

Name	Required	Schema	Description
`YaraRuleContext`	Yes	`YaraRuleContext`	Entity from which YARA rules will be generated.

Responses

POST rules/report request responses

Code	Media type	Response schema	Description
`200`	`application/octet-stream`	`SuccessfulResponse`	OK. The YARA rules were successfully generated. Report format: `yara`.
`202`	`application/json`	`FailureResponse`	YARA rules were not found. The response contains the description and code of an error.
`400`	`application/json`	`ErrorResponse`	Bad request. Format of the specified YARA rule context is incorrect or content of its parameters is invalid.
`401`	`application/json`	`ErrorResponse`	Unauthorized. Authorize using your API key.
`500`	`application/json`	`FailureResponse`	Server error. The response contains the description and code of an error.

Method examples

Python example

```
import requests
def get_yara_report():
    yara_report_url = 'https://ktae-service.com/api/rules/report'
    json = {
        'name': 'exampleName',
        'description': 'exampleDescription',
        'md5s': [
            '34beb7af07a5fc6c89d15385ea926a86',
            '7a2cd5ae4c256bb7a43b652c6fede459'
        ],
        'filter': {
            'startTime': 1620990460532,
            'endTime': 1620991560532,
            'userIds': [
                '92ca7dd8-a10a-11eb-983a-f48e3897cb55',
                '9da1b330-a10a-11eb-8351-f48e3897cb55'
            ]
        },
        'genotypesThreshold': 5,
        'stringsThreshold': 2,
        'sampleTypes': 'initial',
        'simpleRulesEnabled': True,
        'clusterRulesEnabled': False
    }
    headers = {'api-key': '1832611e-7e3e-454d-b51b-7610232a08bb'}

    api_response = requests.post(url=yara_report_url, headers=headers, json=json)
    return api_response
```

cURL example

```
curl -X 'POST' \
  'https://ktae-service.com/api/rules/report' \
  -H 'api-key: 1832611e-7e3e-454d-b51b-7610232a08bb' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "exampleName",
    "description": "exampleDescription",
    "md5s": [
      "34beb7af07a5fc6c89d15385ea926a86",
      "7a2cd5ae4c256bb7a43b652c6fede459"
    ],
    "filter": {
      "startTime": 1620990460532,
      "endTime": 1620991560532,
      "userIds": [
        "92ca7dd8-a10a-11eb-983a-f48e3897cb55",
        "9da1b330-a10a-11eb-8351-f48e3897cb55"
      ]
    },
    "genotypesThreshold": 5,
    "stringsThreshold": 2,
    "sampleTypes": "initial",
    "simpleRulesEnabled": true,
    "clusterRulesEnabled": false
  }'
```

Product state methods

This section describes the product state methods.

GET version

The `version` method receives the system dates and information about Kaspersky Threat Attribution Engine version.

Path

`/api/version`

Request method

`GET`

Request parameters

This request has no parameters.

Responses

GET version request responses

Code	Media type	Response schema	Description
200	application/json	KtaeVersionResponse	OK. The system dates were successfully received.
500	application/json	FailureResponse	Server error. The response contains the description and code of an error.

Method examples

Python example

```
import requests
def get_version():
    version_url = 'https://ktae-service.com/api/version'
    api_response = requests.get(url=version_url)
    return api_response
```

cURL example

```
curl -X 'GET' \
  'https://ktae-service.com/api/version' \
  -H 'accept: application/json'
```

GET healthcheck

The `healthcheck` method receives information about the system health.

Path

`/api/healthcheck`

Request method

`GET`

Request parameters

This request has no parameters.

Responses

GET healthcheck request responses

Code	Media type	Response schema	Description
200	application/json	KtaeStatusResponse	OK. The information about the system health was successfully received.
500	application/json	ErrorResponse	Server error. The response contains the description of an error.

Method examples

Python example

```
import requests
def health_check():
    health_check_url = 'https://ktae-service.com/api/healthcheck'
    api_response = requests.get(url=health_check_url)
    return api_response
```

cURL example

```
curl -X 'GET' \
  'https://ktae-service.com/api/healthcheck' \
  -H 'accept: application/json'
```

GET license

The `license` method receives the information about Kaspersky Threat Attribution Engine license.

Path

`/api/license`

Request method

`GET`

Request parameters

This request has no parameters.

Responses

GET license request responses

Code	Media type	Response schema	Description
200	application/json	LicenseResponse	OK. License information was successfully received.
401	application/json	ErrorResponse	Unauthorized. Authorize using your API key.
500	application/json	FailureResponse	Server error. The response contains the description and code of an error.

Method examples

Python example

```
import requests
def license():
    license_url = 'https://ktae-service.com/api/license'
    headers = {'api-key': '1832611e-7e3e-454d-b51b-7610232a08bb'}
    api_response = requests.get(url=license_url, headers=headers)
    return api_response
```

cURL example

```
curl -X 'GET' \
  'https://ktae-service.com/api/license' \
  -H 'accept: application/json' \
  -H 'api-key: 1832611e-7e3e-454d-b51b-7610232a08bb'
```

Data models

This section describes data models, or schemas, that are available in Kaspersky Threat Attribution Engine REST API.

SuccessfulResponse

The `SuccessfulResponse` data model contains API response messages.

Properties

`SuccessfulResponse` properties

Name	Required	Multiple	Value type	Description
`message`	Yes	No	String	API response message.

Example

```
{
    "message": "Message string"
}
```

ErrorResponse

The `ErrorResponse` data model contains an error description.

Properties

`ErrorResponse` properties

Name	Required	Multiple	Value type	Description
`error`	Yes	No	String	Error description.

Example

```
{
    "error": "Error description"
```

```
}
```

MalformedInputResponse

The `MalformedInputResponse` data model contains descriptions of several errors.

Properties

`MalformedInputResponse` properties

Name	Required	Multiple	Value type	Description
`errors`	No	Yes	`string`	Description of errors.

Example

```
{
    "errors": [
        "The first error description",
        "The second error description"
    ]
}
```

FailureResponse

The `FailureResponse` data model contains a failure description and a code.

Properties

`FailureResponse` properties

Name	Required	Multiple	Value type	Description
`error`	Yes	No	String	Error description.
`code`	Yes	No	Integer	Error code. See the table below for code description.

Error codes

Error code description

Code	Error	Description
`500`	`InternalError`	An internal system error happened.
`501`	`UploadFileError`	An error happened during the file uploading.
`502`	`FileNotFound`	The requested file is not found.
`503`	`MaxFilesLimitExceed`	The maximum number of files to upload is exceeded.
`504`	`OperationDenied`	The requested operation is denied. For example, this error can happen when a regular user tries to access analyses submitted from another user or a user with administrator rights.

505	DatabaseError	An internal database error happened.
506	UnpackPasswordLengthExceeded	Length of a custom password specified to unpack an archive exceeds the limit.
510	AttributionEntityAlreadyExists	The attribution entity with the entered name already exists in the system.
511	AttributionEntityNotFound	The attribution entity with the entered name is not found in the system.
512	ParentAttributionEntityNotFound	The parent attribution entity with the entered name is not found in the system.
513	AttributionEntityNameLengthExceeded	The length of the entered attribution entity name exceeds the limit.
514	AttributionEntityDescriptionLengthExceeded	The length of the entered attribution entity description exceeds the limit.
515	AttributionEntityAliasesLengthExceeded	The length of the attribution entity alias or the number of entered aliases exceed the limit.
516	AttributionEntityReportsLengthExceeded	The length of report links for the attribution entity or the number of entered report links exceed the limit.
517	AttributionEntityCommentsLengthExceeded	The length of comments for the attribution entity exceeds the limit.
518	AttributionEntityThresholdsNegative	The negative value is set for the attribution entity threshold.
531	SampleIsGood	The sample is trusted.
532	SampleHasEmptyData	Samples does not contain any strings or genotypes.
533	SampleHasOnlyGoodData	All strings and genotypes in the sample are trusted.
534	SampleThresholdsError	String and genotype similarity thresholds are set incorrectly for the sample.
540	AnalyzingInProgress	Analysis is in progress.
541	AnalyzeResultsNotFound	The requested analysis result is not found.
542	AnalyzeLimitExceed	The weekly license analysis limit is exceeded.
543	AnalyzingSampleMetaDataNotFound	Metadata for generating YARA rules for the requested sample is not found in the system.
544	AnalyzingIsPending	Analysis request is added to analysis queue and is pending.
545	AnalysisQueueLimitIsExceeded	The limit for the number of queued analyses is exceeded.
546	AnalyzingIsUploading	Analysis request is being uploaded. This status is returned when the samples sent for analysis are uploaded to the processing server.
547	AnalysisNotFound	Analysis with the requested analysis_id is not found.
550	InvalidCredentials	The user is not authorized because the credentials or an API key are invalid.
551	SystemSettingsNotFound	Could not obtain system settings.
560	YaraRulesNotFound	Could not generate YARA rules for the samples.
561	InvalidYaraRuleContext	Data for generating YARA rules is invalid.
562	YaraRuleSamplesLimitExceeded	The maximum number of samples to generate YARA rules is exceeded.
570	HistoryReportLimitExceeded	The maximum number of samples to export from history is exceeded.
600	TimeoutExceeded	The system response timeout is exceeded.

Example

```
{
    "error": "AnalyzingInProgress",
    "code": 540
}
```

AttributionEntity

The `AttributionEntity` data model contains properties of an attribution entity.

Properties

`AttributionEntity` properties

Name	Required	Multiple	Value type	Description
`name`	Yes	No	String	Attribution entity name.
`parent`	No	No	String	Parent attribution entity name.
`description`	No	No	String	Attribution entity description.
`comments`	No	No	String	Attribution entity comment.
`minSimGenotypesThreshold`	No	No	Integer	Minimum value of a genotype threshold among all samples that are attached to the attribution entity. Default value is 2.
`minSimStringsThreshold`	No	No	Integer	Minimum value of a string threshold among all samples that are attached to the attribution entity. Default value is 2.
`reportLinks`	No	No	String	Links to reports about this attribution entity. Multiple report links are separated with a comma.
`aliases`	No	No	String	Known aliases for this attribution entity. Multiple aliases are separated with a comma.

Example

```
{
      "name": "entity",
      "parent": "parent",
      "description": "Entity description",
      "comments": "Comment text",
      "minSimGenotypesThreshold": 2,
      "minSimStringsThreshold": 2,
      "reportLinks": "https://reports.example.com/entity,https://reports.example.com/aliasentity",
      "aliases": "aliasentity"
}
```

AnalysisResponse

The `AnalysisResponse` data model contains analysis ID.

Properties

`AnalysisResponse` properties

Name	Required	Multiple	Value type	Description
`analysisId`	Yes	No	String	Analysis ID in UUID format.

Example

```
{
    "analysisId": "string"
}
```

File

The `File` data model contains file content.

Properties

`File` properties

Name	Required	Multiple	Value type	Description
`data`	Yes	No	Binary	File content in a binary format.

Files

The `Files` data model contains content of multiple files.

Properties

`Files` properties

Name	Required	Multiple	Value type	Description
`data`	Yes	Yes	Binary	Content of multiple files in a binary format

UsedByData

The `UsedByData` data model contains information about attribution entities that use a genotype or a string from an analyzed sample.

Properties

`UsedByData` properties

Name	Required	Multiple	Value type	Description
`attribution_entity`	Yes	No	String	Attribution entity name.
`matched`	Yes	No	Integer	Number of attribution entity samples that have this genotype or string.

Example

```
{
    "attribution_entity": "entity1",
```

```
    "matched": 2
}
```

Top5Data

The `Top5Data` data model contains top five attribution entities with samples that are the most similar to the analyzed sample.

Properties

`Top5Data` properties

Name	Required	Multiple	Value type	Description
`attribution_entity`	Yes	No	String	Attribution entity name.
`similarity`	Yes	No	Integer	Similarity percentage.

Example

```
{
    "attribution_entity": "entity1",
    "similarity": 100
}
```

SimilarData

The `SimilarData` data model contains information about an attribution entity sample that is similar to an analyzed sample.

Properties

`SimilarData` properties

Name	Required	Multiple	Value type	Description
`md5`	Yes	No	String	MD5 hash of the similar sample.
`size`	Yes	No	Integer	Size of the similar sample, in bytes.
`matched_genotypes`	Yes	No	Integer	Number of matching genotypes between the analyzed sample and the similar attribution entity sample.
`total_genotypes`	Yes	No	Integer	Total number of genotypes in the similar attribution entity sample.
`matched_strings`	Yes	No	Integer	Number of matching strings between the analyzed sample and the similar attribution entity sample.
`total_strings`	Yes	No	Integer	Total number of strings in the similar attribution entity sample.
`similarity`	Yes	No	Integer	Similarity percentage between the analyzed sample and the similar attribution entity sample.
`attribution_entity`	Yes	No	String	Name of the attribution entity that owns the similar sample.
`aliases`	Yes	Yes	String	Sequence of strings, separated by comma, that includes aliases of the attribution entity that owns the sample.

Example

```
{
    "md5": "44d88612fea8a8f36de82e1278abb02f",
    "size": 128,
    "matched_genotypes": 1
    "total_genotypes": 1,
    "matched_strings": 1,
    "total_strings": 1,
    "similarity": 100,
    "attribution_entity": "eicar",
    "aliases": "alias1", "alias2"
}
```

GenotypeData

The `GenotypeData` data model contains information about a genotype in the analyzed sample that matched to genotypes of similar samples.

Properties

`GenotypeData` properties

Name	Required	Multiple	Value type	Description
`genotype`	Yes	No	String	Genotype in the analyzed sample that matched to genotypes of similar samples.
`matched`	Yes	No	Integer	Number of known attribution entity samples that have this genotype.
`used_by`	Yes	Yes	`UsedByData`	Array of attribution entities that own samples with this genotype.

Example

```
{
    "genotype": "0c91483bf75cab06bca6dbd41ff2a54e",
    "matched": 1,
    "used_by": [
        {
            "attribution_entity": "entity1",
            "matched": 1
        }
    ]
}
```

StringData

The `StringData` data model contains information about a string in the analyzed sample that matched to strings in similar samples.

Properties

`StringData` properties

Name	Required	Multiple	Value type	Description
`string`	Yes	No	String	String in the analyzed sample that matched to strings of similar samples.

matched	Yes	No	Integer	Number of known attribution entity samples that have this string.
used_by	Yes	Yes	UsedByData	Array of attribution entities that own samples with this string.

Example

```
{
  "string": "ExampleString",
  "matched": 1,
  "used_by": [
    {
      "attribution_entity": "entity1",
      "matched": 1
    }
  ]
}
```

PreviouslyData

The `PreviouslyData` data model contains information about previously analyzed samples that are similar to the analyzed sample.

Properties

`PreviouslyData` properties

Name	Required	Multiple	Value type	Description
md5	Yes	No	String	MD5 hash of the previously analyzed sample.
filename	Yes	No	String	Name of the previously analyzed sample.
size	Yes	No	Integer	Size of the previously analyzed sample, in bytes.
matched_genotypes	Yes	No	Integer	Number of matching genotypes between the analyzed sample and the similar previously analyzed sample.
total_genotypes	Yes	No	Integer	Total number of genotypes in the similar previously analyzed sample.
matched_strings	Yes	No	Integer	Number of matching strings between the current sample and similar previously analyzed sample.
total_strings	Yes	No	Integer	Total number of strings in the similar previously analyzed sample.
similarity	Yes	No	Integer	Similarity percentage between the analyzed sample and the previously analyzed sample.
top_5	Yes	Yes	Top5Data	Array with the top five attribution entity objects that have samples with the highest similarity to the previously analyzed sample.
last_analysis_date	Yes	No	String	Last analysis date of the previously analyzed sample.
username	Yes	No	String	Name of the user that started the analysis.

Example

```
{
    "md5": "56beb7af07a5fc6c89d15385ea926a86",
    "filename": "exampleFilename",
    "size": 678,
    "matched_genotypes": 2,
    "total_genotypes": 1,
    "matched_strings": 3,
```

```
    "total_strings": 4,
    "similarity": 12,
    "top_5": [
        {
            "attribution_entity": "entity1",
            "similarity": 98
        }
    ],
    "last_analysis_date": "2021-06-23 21:32:04",
    "username": "user name"
}
```

PreviouslyGenotypeData

The `PreviouslyGenotypeData` data model contains statistics about genotypes that matched between the analyzed sample and previously analyzed samples.

Properties

`PreviouslyGenotypeData` properties

Name	Required	Multiple	Value type	Description
`genotype`	Yes	No	String	Genotype of the analyzed sample that matched with genotypes of previously analyzed samples.
`matched`	Yes	No	Integer	Number of previously analyzed samples that contain the genotype.
`used_by`	Yes	Yes	`UsedByData`	Array of attribution entities that own samples with this genotype.
`filenames`	Yes	Yes	String	Comma-separated array that includes names of the files in which the genotype was detected.

Example

```
{
    "genotype": "34beb7af07a5fc6c89d15385ea926a86",
    "matched": 10,
    "used_by": [
        {
            "attribution_entity": "entity2",
            "matched": 2
        }
    ],
    "filenames": [
        "filename1",
        "filename2"
    ]
}
```

PreviouslyStringData

The `PreviouslyStringData` data model contains statistics about strings that matched between the analyzed sample and previously analyzed samples.

Properties

`PreviouslyStringData` properties

Name	Required	Multiple	Value type	Description
string	Yes	No	String	String of the analyzed sample that matched with strings of previously analyzed samples.
matched	Yes	No	Integer	Number of previously analyzed samples that contain the string.
used_by	Yes	Yes	UsedByData	Array of attribution entities that own samples with this string.
filenames	Yes	Yes	String	Comma-separated array that includes names of the files in which the string was detected.

Example

```
{
   "string": "exampleString",
   "matched": 10,
   "used_by": [
       {
           "attribution_entity": "entity2",
           "matched": 2
       }
   ],
   "filenames": [
       "filename1",
       "filename2"
   ]
}
```

ReportData

The ReportData data model contains results of a sample analysis. If sample unpacking was enabled, the response contains an array of ReportData items.

Properties

ReportData properties

Name	Required	Multiple	Value type	Description
md5	Yes	No	String	MD5 hash of the analyzed file.
filename	Yes	No	String	Name of the analyzed file.
size	Yes	No	Integer	Size of the analyzed file, in bytes.
matched_bad_genotypes	Yes	No	Integer	Number of genotypes in the analyzed sample that matched the genotypes in the similar attribution entity samples.
total_bad_genotypes	Yes	No	Integer	Total number of genotypes in the analyzed sample that matched genotypes of all attribution entity samples if similarity threshold is ignored.
matched_bad_strings	Yes	No	Integer	Number of strings in the analyzed sample

				that matched the strings in the similar attribution entity samples.
total_bad_strings	Yes	No	Integer	Total number of strings in the analyzed sample that matched strings of all attribution entity samples if similarity threshold is ignored.
top_5	Yes	Yes	Top5Data	Array with the top five attribution entities that have samples with the highest similarity to the analyzed sample.
previously_matched_samples_count	Yes	No	Integer	Number of similar previously analyzed samples.
detection_names	Yes	Yes	String	Comma-separated array with names of detections that are found by anti-virus databases.
analysis_date	Yes	No	String	Analysis date and time, in UTC format.
username	Yes	No	String	Name of the user that created a report request.
result_code	Yes	No	Integer	Analysis result code. For a list of result codes, see About result and unpacking codes.
unpacking_status	Yes	No	Integer	Unpacking operation status. For a list of unpacking status codes, see About result and unpacking codes.
parent_md5	Yes	No	String	MD5 hash of the parent file of the analyzed file. This parameter is shown for unpacked archives.
extracted_path	Yes	No	String	Path to the analyzed file relative to its parent file. This parameter is shown for unpacked archives.
similar_attribution_entities_samples	Yes	Yes	SimilarData	Array with similar attribution entity samples.
matched_genotypes_with_attribution_entity_samples	Yes	Yes	GenotypeData	Array of genotypes that matched between the analyzed sample and a similar attribution entity sample.
matched_strings_with_attribution_entity_samples	Yes	Yes	StringData	Array of strings that matched between

				the analyzed sample and a similar attribution entity sample.
similar_previously_analyzed_samples	Yes	Yes	PreviouslyData	Statistics for previously analyzed samples that are similar to the analyzed sample.
matched_genotypes_with_previously_analyzed_samples	Yes	Yes	PreviouslyGenotypeData	Statistics for genotypes that matched between the analyzed sample and similar previously analyzed samples.
matched_strings_with_previously_analyzed_samples	Yes	Yes	PreviouslyStringData	Statistics for strings that matched between the analyzed sample and similar previously analyzed samples.

Example


```
{
    "md5": "b8b10c05d6b8af8f7f9a6a4bfd5bf8fe",
    "filename": "example.exe",
    "size": 245872,
    "total_bad_genotypes": 9,
    "matched_bad_genotypes": 5,
    "total_bad_strings": 7,
    "matched_bad_strings": 2,
    "top_5": [
        {
            "attribution_entity": "entity0",
            "similarity": 97
        },
        {
            "attribution_entity": "entity1",
            "similarity": 12
        }
    ],
    "previously_matched_samples_count": 1,
    "detection_names": [
        "detectionName1",
        "detectionName2"
    ],
    "analysis_date": "2021-04-18 13:05:02",
    "username": "exampleUsername",
    "result_code": 2,
    "unpacking_status": 0,
    "parent_md5": "c2b027626254f5c07518b0be1087a845",
    "extracted_path": "/example.zip/example.exe",
    "similar_attribution_entities_samples": [
        {
            "md5": "f46b8f540a48c80aa470d4e7c6f34e1a",
            "size": 2048,
            "total_genotypes": 3,
            "matched_genotypes": 2,
            "total_strings": 4,
            "matched_strings": 4,
            "similarity": 75,
            "attribution_entity": "entity0",
            "aliases": [
                "exampleAlias1",
                "exampleAlias2"
            ]
        }
    ],
    "matched_genotypes_with_attribution_entity_samples": [
        {
```

```
            "genotype": "66beb7af07a5fc6c89d15385ea926a86",
            "matched": 5,
            "bad_rating": 6,
            "used_by": [
                {
                    "attribution_entity": "entity0",
                    "matched": 1
                },
                {
                    "attribution_entity": "entity1",
                    "matched": 4
                }
            ]
        }
    ],
    "matched_strings_with_attribution_entity_samples": [
        {
            "string": "exampleString",
            "matched": 3,
            "bad_rating": 3,
            "used_by": [
                {
                    "attribution_entity": "entity1",
                    "matched": 2
                },
                {
                    "attribution_entity": "entity0",
                    "matched": 1
                }
            ]
        }
    ],
    "similar_previously_analyzed_samples": [
        {
            "md5": "56beb7af07a5fc6c89d15385ea926a86",
            "filename": "exampleFilename",
            "size": 678,
            "total_genotypes": 7,
            "matched_genotypes": 2,
            "total_strings": 4,
            "matched_strings": 3,
            "similarity": 12,
            "top_5": [
                {
                    "attribution_entity": "entity2",
                    "similarity": 32
                },
                {
                    "attribution_entity": "entity0",
                    "similarity": 12
                }
            ],
            "last_analysis_date": "2021-03-23 21:32:04",
            "username": "exampleUsername"
        }
    ],
    "matched_genotypes_with_previously_analyzed_samples": [
        {
            "genotype": "34beb7af07a5fc6c89d15385ea926a86",
            "matched": 10,
            "bad_rating": 12,
            "used_by": [
                {
                    "attribution_entity": "entity2",
                    "matched": 2
                },
                {
                    "attribution_entity": "entity0",
                    "matched": 8
                }
            ],
            "filenames": [
                "filename1",
                "filename2"
            ]
        }
    ],
    "matched_strings_with_previously_analyzed_samples": [
```

```
        {
            "string": "exampleString",
            "matched": 9,
            "bad_rating": 12,
            "used_by": [
                {
                    "attribution_entity": "entity2",
                    "matched": 1
                },
                {
                    "attribution_entity": "entity0",
                    "matched": 8
                }
            ],
            "filenames": [
                "filename1",
                "filename2"
            ]
        }
    ]
}
```

YaraRuleFilter

The `YaraRuleFilter` data model contains a filter for generating YARA rules.

Properties

`YaraRuleFilter` properties

Name	Required	Multiple	Value type	Description
`startTime`	No	No	Integer	Time, in milliseconds from January 1, 1970, 00:00:00 UTC, from which analyses are included to generate YARA rules.
`endTime`	No	No	Integer	Time, in milliseconds from January 1, 1970, 00:00:00 UTC, up to which analyses are included to generate YARA rules.
`userIds`	No	Yes	String	IDs, in UUID format, of users whose analyses are included to generate YARA rules. This filter is available only for users with administrator rights. A regular user can only generate the YARA rules from the analyses that the user submitted.

Example

```
{
    "startTime": 1620990460532,
    "endTime": 1620991560532,
    "userIds": [
        "2e46cca7-111f-4306-8efb-9795b90d536e",
        "6fcb7bab-1f74-4b36-abd8-04c45759c2c1"
    ]
}
```

YaraRuleContext

The `YaraRuleContext` data model contains entities that are used to generate YARA rules.

Properties

`YaraRuleContext` properties

Name	Required	Multiple	Value type	Description
`name`	Yes	No	String	Name of the YARA rule.
`description`	No	No	String	Description of the YARA rule.
`md5s`	No	Yes	String	MD5 hashes of samples that were used to generate YARA rules. If the field is in the request, YARA rules are generated only for the samples in the field. The `filter` and `sampleTypes` fields are ignored.
`filter`	No	No	`YaraRuleFilter`	Filter used to generate YARA rules. • If this field is not present in the request and the `md5s` field is present in the request, YARA rules are generated only for samples specified in the `md5s` field. The value in the `sampleTypes` field is ignored. • If this field and the `md5s` field are not present in the request, YARA rules are generated only for the samples that were used in the last analysis performed by the user.
`genotypesThreshold`	No	No	Integer	Minimum number of genotypes in the group of samples that must match to generate a cluster YARA rule. Default value is `10`.
`stringsThreshold`	No	No	Integer	Minimum number of strings in the group of samples that must match to generate a cluster YARA rule. Default value is `10`.
`sampleTypes`	No	No	String	Samples to generate YARA rules. This field is used only if the `md5` and `filter` fields are not present in the request. Possible values: • `initial`—Use only initial samples and ignore child samples. • `all`—Use all samples, including child samples. Default value is `initial`.
`simpleRulesEnabled`	No	No	Boolean	Flag to generate simple YARA rules. Default value is `false`. If the `false` value is set in the field, the `true` value must be set in the `clusterRulesEnabled` field.
`clusterRulesEnabled`	No	No	Boolean	Flag to generate cluster YARA rules. Default value is `false`. If the `false` value is set in the field, the `true` value must be set in the `simpleRulesEnabled` field.

Example

```
{
    "name": "exampleName",
    "description": "exampleDescription",
    "md5s": [
        "34beb7af07a5fc6c89d15385ea926a86",
        "45b0aebebcddcb1ff40dc76fbf9937ee"
    ],
    "filter": {
        "startTime": 1620990460532,
        "endTime": 1620991560532,
        "userIds": [
            "2e46cca7-111f-4306-8efb-9795b90d536e",
            "6fcb7bab-1f74-4b36-abd8-04c45759c2c1"
        ]
    },
    "genotypesThreshold": 5,
```

```
    "stringsThreshold": 2,
    "sampleTypes": "initial",
    "simpleRulesEnabled": true,
    "clusterRulesEnabled": true
}
```

KtaeStatusResponse

The `KtaeStatusResponse` data model contains information about Kaspersky Threat Attribution Engine health.

Properties

`KtaeStatusResponse` properties

Name	Required	Multiple	Value type	Description
`databaseStatus`	Yes	No	Boolean	Database operating status. Possible values: • `true` —Database is working properly. • `false` —Database is inaccessible or is not working properly.
`scannerStatus`	Yes	No	Boolean	Scanner operational status. The scanner is a utility used for unpacking the compound objects and providing anti-virus decisions for the analyzed files. Possible values: • `true` —Scanner is working properly. • `false` —Scanner is inaccessible or is not working properly.
`KTAEStatus`	Yes	No	Boolean	Kaspersky Threat Attribution Engine operating status. Possible values: • `true` —Kaspersky Threat Attribution Engine and all its subsystems are working properly. • `false` —Kaspersky Threat Attribution Engine and at least one of its subsystems are inaccessible or are not working properly.
`cipherStatus`	Yes	No	Boolean	Encryptor operating status. Possible values: • `true` —Encryptor is working properly. • `false` —Encryptor is inaccessible or is not working properly.

Example

```
{
    "databaseStatus": true,
    "scannerStatus": true,
    "KTAEStatus": true,
    "cipherStatus": true
}
```

KtaeVersionResponse

The `KtaeVersionResponse` data model contains system dates and information about the current Kaspersky Threat Attribution Engine version.

Properties

`KtaeVersionResponse` properties

Name	Required	Multiple	Value type	Description
`version`	Yes	No	String	Kaspersky Threat Attribution Engine version.
`licenseExpirationDate`	Yes	No	String	License expiration date.
`ktaeBasesDate`	Yes	No	String	Date when the Kaspersky Threat Attribution Engine databases were last updated.
`AVBasesDate`	Yes	No	String	Date when the anti-virus databases used by the application were uploaded to the update server.

Example

```
{
    "version": "1.0.0",
    "licenseExpirationDate": "2021-04-10T11:20:00Z",
    "ktaeBasesDate": "2021-04-12T10:20:00Z",
    "AVBasesDate": "2021-04-15T12:20:00Z"
}
```

HistoryReportFilter

The `HistoryReportFilter` data model contains a filter for generation reports from analysis history.

Properties

`HistoryReportFilter` properties

Name	Required	Multiple	Value type	Description
`userIds`	No	Yes	String	IDs, in UUID format, of users whose analyses are included to the report. This filter is available only for users with administrator rights. A regular user can only generate reports from the analyses that the user submitted.
`startTime`	No	No	Integer	Time, in milliseconds from January 1, 1970, 00:00:00 UTC, from which analyses are included to the report.
`endTime`	No	No	Integer	Time, in milliseconds from January 1, 1970, 00:00:00 UTC, up to which analyses are included to the report.
`minSimilarity`	No	No	Integer	Minimum similarity threshold for the attribution entity samples.
`name`	No	No	String	Name of the report file.

Example

```
{
    "userIds": [
        "2e46cca7-111f-4306-8efb-9795b90d536e",
        "6fcb7bab-1f74-4b36-abd8-04c45759c2c1"
    ],
    "startTime": 1620990460532,
    "endTime": 1620991560532,
    "minSimilarity": 5,
    "name": "reportName"
}
```

ReportLimitExceed

The `ReportLimitExceed` data model contains description of the error that appears when the maximum number of samples to export from history is exceeded.

Properties

`ReportLimitExceed` properties

Name	Required	Multiple	Value type	Description
`current`	Yes	No	Integer	Number of samples that were selected for export.
`limit`	Yes	No	Integer	Maximum number of samples that can be exported.

Example

```
{
     "current": 1200,
     "limit": 1000
}
```

LicenseResponse

The `LicenseResponse` data model contains information about Kaspersky Threat Attribution Engine license.

Properties

`LicenseResponse` properties

Name	Required	Multiple	Value type	Description
`licenseName`	Yes	No	String	Kaspersky Threat Attribution Engine license name.
`licenseExpirationDate`	Yes	No	String	License expiration date.
`licenseRemainingDays`	Yes	No	Integer	Number of days before license expiration date.
`licenseAvailableAnalyzeCount`	No	No	Integer	Number of analyses that are available for a week according to the

				license.
`licenseRemainingAnalyzeCount`	No	No	Integer	Number of analyses that remain for a week according to the license.

Example

```
{
    "licenseName": "Kaspersky Threat Attribution Engine, On-premise edition",
    "licenseExpirationDate": "2022-04-10 11:20:00",
    "licenseRemainingDays": 213,
    "licenseAvailableAnalyzeCount": 100,
    "licenseRemainingAnalyzeCount": 19
}
```

Limitations and warnings

Limitation for analyzing files in a batch

If a batch of files is submitted, similarities between them are not highlighted.

Limitation for sending files using API

If samples are sent for analysis using API, similar samples are not searched among the previously analyzed samples by default. In the **Previously matched samples** column of the web interface, the — value is displayed for such files.

Limitation for analysis history storage

The storage period for analysis history is constant and equals to 180 days. Currently, this setting cannot be changed from the web interface.

Contacting Technical Support

If you cannot find a solution to your problem in the Kaspersky Threat Attribution Engine documentation, contact Technical Support.

Kaspersky provides the customers with Technical Support, which includes resolving customer's problems related to the purchased product. Kaspersky makes sure that the customer uses a product properly according to the product intended use and in compliance with the documentation and technical specifications.

Kaspersky provides support for Kaspersky Threat Attribution Engine during its lifecycle (see the product support lifecycle page). Before contacting Technical Support, please read the support rules.

The customer should provide as much detailed information as possible to help Kaspersky Technical Support resolve the problem or to enable Kaspersky to fix any issues in the product or further develop the product as may be required by Kaspersky.

Information about third-party code

Information about third-party code is contained in the following files:

- docs/legal_notices.txt file in the /opt/ktae directory
- docs/legal_notices.txt file in the /opt/ktae-updater directory

Trademark notices

Registered trademarks and service marks are the property of their respective owners.

Amazon, Amazon Web Services, and AWS are trademarks of Amazon.com, Inc. or its affiliates.

Safari is a trademark of Apple Inc.

Chromium, and Google Chrome are trademarks of Google LLC.

Intel and Xeon are trademarks of Intel Corporation or its subsidiaries.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Microsoft, and Microsoft Edge are trademarks of the Microsoft group of companies.

Mozilla and Firefox are trademarks of the Mozilla Foundation in the U.S. and other countries.

OpenSSL is a trademark owned by the OpenSSL Software Foundation.

Java is a registered trademark of Oracle and/or its affiliates.

Python is a trademark or registered trademark of the Python Software Foundation.

CentOS is a trademark or registered trademark of Red Hat, Inc. or its subsidiaries in the United States and other countries.

Rocky Linux is a trademark of The Rocky Enterprise Software Foundation.