kaspersky

Kaspersky Threat Feed App for MISP

Product version 1.1

kaspersky

Dear User,

Thank you for choosing Kaspersky as your security software provider. We hope that this document helps you to use our product.

Document revision date: 11/1/2019

https://www.kaspersky.com
https://help.kaspersky.com
https://support.kaspersky.com

kaspersky

Contents

kaspersky

About this document

This document describes Kaspersky Threat Feed App for MISP, a utility developed by Kaspersky that imports Kaspersky Threat Data Feeds to a Malware Information Sharing Platform (MISP) instance.

kaspersky

About MISP

Malware Information Sharing Platform (MISP) is an open-source software solution for collecting, storing, distributing, sharing, and correlating Indicators of Compromise. There can be Indicators of Compromise of targeted attacks, threat intelligence, financial fraud information, vulnerability information, or even counter-terrorism information. The objective of MISP is to foster the sharing of structured information within the security community. MISP provides functionalities to support exchange of information but also consumption of the information by Intrusion Detection Systems (IDS), log analysis tools, and SIEM software.

The MISP features include the following:

- Importing indicators from MISP, STIX™, OpenIOC, text, and CSV data
- Automatic information sharing about threats among various participants

  There are a number of open MISP communities in which you can participate.
- Automatic generating rules for IDS Bro, Snort®, and Suricata, and for various SIEM software programs

MISP includes many Python® modules for integration with various software programs:

- Expansion modules—Modules that enrich events with some data.

  Expansion modules can be of two types:

  - Hover type

    Modules that display enriched events without modifying the events.
  - Expansion type

    Modules that modify events by enriching them with data and displaying the result.
- Import modules—Modules that import indicators to MISP.
- Export modules—Modules that export data from MISP (for example, to SIEM software).

kaspersky

About Kaspersky Threat Feed App for MISP

Kaspersky Threat Feed App for MISP imports and updates Kaspersky Threat Data Feeds in a MISP instance. Every feed is imported as a MISP event. Indicators from the feeds are added to events as attributes. The supported feeds are described in section "Feeds from Kaspersky (on page 7)".

In this chapter

Distribution kit

Kaspersky Threat Feed App for MISP is shipped as an archive. The contents of the archive are described in the following table.

Table 1. Package contents

Item	Description
feed_util/*	Directory that contains Kaspersky Feed Utility.
defs.py	Service file for the importing utility and run.py script.
import_to_misp.py	Script that imports feeds to a MISP instance.
Kaspersky Threat Feed App for MISP.pdf	Kaspersky Threat Feed App for MISP documentation.
kl_feeds_converter.py	Service file for the importing utility.
legal_notices.txt	Legal notices for the product.
license.txt	End User License Agreement (EULA).
misp_api.py	Service file for the importing utility.
process_feed.py	Service file for the importing utility.
requirements.txt	List of Python packages necessary for operation of the utility.
rollback.py	Service file for the script run.py.
run.py	Script that runs, successively, Feed Utility and the importing script.
settings.py	Script with settings for the script run.py.
utils.py	Service file for the importing utility.

kaspersky

Hardware and software requirements

Kaspersky Threat Feed App for MISP has the following system requirements.

Supported operating systems

Kaspersky Threat Feed App for MISP can run on Linux® x64.

Software requirements

Kaspersky Threat Feed App for MISP requires Python 3.6 or later.

The current version of Kaspersky Threat Feed App for MISP proved to work on MISP v2.4.116.

RAM requirements

Kaspersky Threat Feed App for MISP uses up to 2 gigabytes (GB) of RAM when importing feeds to MISP.

Feeds from Kaspersky

This section describes Kaspersky Threat Data Feeds that can be imported to a MISP instance.

The following feeds are available:

- Malicious URL Exact Data Feed—A set of exact URLs and FQDNs with context that refer to malicious websites and web pages.
- Phishing URL Exact Data Feed—A set of exact URLs and FQDNs with context that refer to phishing websites and web pages.
- Botnet CnC URL Exact Data Feed—A set of exact URLs, exact FQDNs, and hashes with context that refer to desktop botnet C&C servers and related malicious objects.
- Malicious Hash Data Feed—A set of file hashes with corresponding context covering the most dangerous, prevalent, or emerging malware.
- Mobile Malicious Hash Data Feed—A set of file hashes with corresponding context for detecting malicious objects that infect mobile Google Android and Apple iPhone devices.
- P-SMS Trojan Data Feed—A set of Trojan hashes with corresponding context for detecting SMS Trojans that send premium-rate SMS messages to mobile users as well as enable attackers to steal, delete, and respond to SMS messages.
- IP Reputation Data Feed—A set of IP addresses with context covering spam hosts, malicious hosts, phishing hosts, Tor exit nodes, proxies, and botnet C&C servers.
- Mobile Botnet Data Feed—A set of URLs and hashes with context covering mobile botnet C&C servers.
- Ransomware URL Feed—A set of URLs with corresponding context for detecting links and websites that host ransomware.

Demo feeds are also available. Demo feeds provide lower detection rates in comparison with their corresponding commercial versions. The following demo feeds are available:

- Demo Botnet CnC URL Data Feed

  This is a demo version of Botnet CnC URL Data Feed.
- Demo Malicious Hash Data Feed

This is a demo version of Malicious Hash Data Feed.

- Demo IP Reputation Data Feed

  This is a demo version of IP Reputation Data Feed.

kaspersky

Using Kaspersky Threat Feed App for MISP

This section describes how to use Kaspersky Threat Feed App for MISP.

In this chapter

Installing Kaspersky Threat Feed App for MISP

This section describes how to install Kaspersky Threat Feed App for MISP.

► *To install Kaspersky Threat Feed App for MISP:*

1. Unpack the distribution kit contents to the directory you want.

   Hereinafter, this directory is referred to as `%utility_dir%`.

   > We recommend that you assign permissions to gain access to the directory and all its parent directories so that no other user can modify these directories.

2. Install the libraries listed in the requirements.txt file that are not present on the computer.

   Do this by running the following command:

   ```
   pip install -r %utility_dir%/requirements.txt
   ```

   Depending on the configuration of your operating system, the Python package installer can use a different command to install modules. For example, `pip3`.

3. Read the End User License Agreement (EULA). You can find the terms of the EULA in the `%utility_dir%/license.txt` file.

   If you agree to the terms of the EULA, proceed to the next step. If you do not agree to the terms of the EULA, cancel the installation.

4. Accept the EULA by changing the value of the `<EULA>` element in the template.conf file to `<EULA>accepted</EULA>`.

   Kaspersky Feed Utility runs only if the EULA is accepted.

5. Save and close the template.conf file.

6. Configure Kaspersky Threat Feed App for MISP (see section "Configuring Kaspersky Threat Feed App for

MISP" on page 10).

7. Add the following line to the crontab file:

```
0 0 * * * %utility_dir%/run.py
```

Subsequently, the master script run.py will be run regularly (every 24 hours): Feed Utility will download the feeds from the Kaspersky servers, and then the importing script import_to_misp.py will import the feeds to a MISP instance.

Configuring Kaspersky Threat Feed App for MISP

You can configure or reconfigure Kaspersky Threat Feed App for MISP mainly by editing the settings script file settings.py.

► *To configure Kaspersky Threat Feed App for MISP:*

1. In the settings script file settings.py, set the following settings:

   - `MISP_URL`

     This is the URL where the MISP instance is available.

     The run.py script adds this value after the `--misp_url` option of the import_to_misp.py script (see section "Command-line options" on page 12).

   - `MISP_TOKEN`

     This is the key for using the MISP instance.

     The run.py script adds this value after the `--auth_key` option of the import_to_misp.py script (see section "Command-line options" on page 12).

   - `CERT_NO_VERIFY`

     This setting enables (when its value is `False`) or disables (when its value is `True`) the SSL certificate verification when the MISP API is used.

     If this setting is `True`, the run.py script uses the `--no_verification` option of the import_to_misp.py script (see section "Command-line options" on page 12).

     This parameter is intended only for evaluation purposes. Using this parameter in a production environment may create security issues.

   - `LOG_LEVEL`

     Defines the logging level for a converter.

     Two logging levels are available: `DEBUG` and `INFO`. The `DEBUG` level is used by default.

     When the `INFO` logging level is enabled, the converter writes less information to the log files than when the `DEBUG` level is enabled. Before enabling the `INFO` debug level, make sure that the converter works without errors. Otherwise, the information about errors may not be logged or may be logged only partially on the `INFO` debug level.

- LOG_OUTPUT

  Defines the output format for logs. Logs can be written to a file or to the stdout stream.

  Feed Utility logging settings are not affected by this parameter and can be configured by changing the LogSettings parameter in the %utility_dir%/feed_util/template.conf file. For more information about Feed Utility logging, refer to the online documentation for Kaspersky CyberTrace, web page https://click.kaspersky.com/?hl=en-US&link=online_help&pid=CyberTrace&version=1.0&helpid=171650.

  This parameter can have the following values: STDOUT, FILE. The default value for this parameter is STDOUT.

- LOG_FILENAME

  Defines a path and file name for the log file.

  If a path is not specified, the log file is created in the %utility_dir% directory. Make sure that a user that runs the converter has sufficient rights to write to this file.

- RECORDS_COUNT

  This setting specifies the maximum number of attributes that a MISP event will contain after a feed is imported to a MISP instance. It must be a non-negative integer (0 means no limit). The default value is 200000.

  The run.py script adds this value after the --attributes_limit option of the import_to_misp.py script (see section "Command-line options" on page 12).

2. In the settings script file settings.py, in the FEEDS dictionary, uncomment the lines containing the names of the feeds that you will import to a MISP instance.

   > We recommend that you not use commercial feeds together with their demo versions. If you plan to use commercial feeds after you have used demo feeds, remove the MISP events that correspond to demo feeds.

   Also, you can edit the filters to be applied to feed records. The filter rules are defined in the filters subdictionary and re-used by Feed Utility without modification. For more information about filtering rules used by Feed Utility, visit https://click.kaspersky.com/?hl=en-US&link=online_help&pid=CyberTrace&version=1.0&helpid=171652.

   By default, the following filter is set for the records of IP Reputation Data Feed and Demo IP Reputation Data Feed:

   ```
   'Demo_IP_Reputation_Data_Feed': {'id': '87', 'filters': {'threat_score': '[75;*]'}},

   'IP_Reputation_Data_Feed': {'id': '68', 'filters': {'threat_score': '[75;*]'}},
   ```

   The default filter allows importing only those feed records that contain IP addresses considered dangerous (namely, the threat_score field for which is equal to or exceeds 75).

3. Copy the certificate for moving feeds to the same directory where the Feed Utility binary file resides and rename the certificate file to feeds.pem.

4. If you are going to use a proxy server to gain access to Kaspersky servers to download feeds, from the

Feed Utility directory run the following command once:

```
kl_feed_util --set-proxy %PROXY% -c template.conf
```

Here `%PROXY%` is the setting that specifies the proxy server parameters for gaining access to the Kaspersky servers.

Recommendations on the number of attributes kept in a MISP event

Updating the full set of Kaspersky Threat Data Feeds in a MISP instance can take a significant amount of time. You can decrease this time by specifying the maximum number of attributes to keep in a MISP event (the `RECORDS_COUNT` parameter in the settings.py file). We recommend that you set the maximum number of attributes to 200 000 (this value is initially set in the settings.py file).

In the table below, measurement data is provided that indicates the influence of the maximum number of attributes in a MISP event on the import time. This measurement data is got when all feeds were imported on our computers. Real data (other measurement data) depends on the software and hardware that you use, on the period between subsequent running of the importing script, and on the feeds you import to a MISP instance.

Table 2. Import time depending on the maximum number of attributes

Maximum number of attributes	Time of the first import, hours	Time of a regular import, hours
50 000	3	2
200 000	13	4
500 000	24	8

Removing Kaspersky Threat Feed App for MISP

This section describes how to remove Kaspersky Threat Feed App for MISP.

► *To remove Kaspersky Threat Feed App for MISP:*

1. From the crontab file, remove the line that corresponds to running the master script file run.py.
2. Wait until Kaspersky Threat Feed App for MISP finishes operating or stop execution of its process manually.
3. Remove the `%utility_dir%` directory and its contents.

Command-line options

Kaspersky Threat Feed App for MISP is designed in such a way that the user periodically runs the master script file run.py. However you can run the importing script import_to_misp.py instead. The import_to_misp.py script runs from the command line as follows:

```
python %utility_dir%/import_to_misp.py --misp_url <MISP_URL> --auth_key
<MISP_authorization_key> --proxy <proxy_to_MISP> --feed_file <feed> --deleted_file
<file_with_deleted_records> --added_file <file_with_added_records> --work_dir
<working_directory> [-nv]
```

On your computer, the command that runs Python may have a different name (for example, `python3` or `py`).

The following table contains the description of the command-line parameters.

Table 3. Parameters of the importing script

Parameter	Description
`--misp_url`	URL or IP address at which the MISP instance is available. If your MISP instance uses an SSL certificate to establish secure connections over HTTPS, then the URL must begin with the `https://` protocol specifier. Otherwise, the converter will not be able to add, modify, or delete events and attributes in MISP during working with the MISP API. The `misp_url` parameter is mandatory.
`--auth_key`	Key for gaining access to the MISP instance. The `auth_key` parameter is mandatory.
`--proxy`	Proxy settings in the `http://username:password@address:port` format. This is the proxy server for gaining access to the MISP instance. The `proxy` parameter is optional. If the parameter is not specified, no proxy server is used.
`--feed_file`	Path to the feed file to be initially imported. It can be an absolute or relative path. A relative path is calculated relative to the import_to_misp.py file. Only local paths are supported; SMB, FTP, or HTTP paths are not supported. The `feed_file` parameter must be specified if the `deleted_file` and `added_file` parameters are not specified. Otherwise, it must not be specified.
`--deleted_file`	Path to the diff feed file that contains deleted records. It can be an absolute or relative path. A relative path is calculated relative to the import_to_misp.py file. Only local paths are supported; SMB, FTP, or HTTP paths are not supported. The `deleted_file` parameter must be specified if the `added_file` parameter is specified. Otherwise, it must not be specified.
`--added_file`	Path to the diff file that contains added records. It can be an absolute or relative path. A relative path is calculated relative to the import_to_misp.py file. Only local paths are supported; SMB, FTP, or HTTP paths are not supported. The `added_file` parameter must be specified if the `deleted_file` parameter is specified. Otherwise, it must not be specified.
`--work_dir`	Path to the working directory. It can be an absolute or relative path. A relative path is calculated relative to the import_to_misp.py file. Only local paths are supported; SMB, FTP, or HTTP paths are not supported. To prevent the feed file from being overwritten with a temporary file, do not specify the directory containing the feed file (see the description of the `feed_file` parameter) in the `work_dir` parameter. The `work_dir` parameter is mandatory.

`--attributes_limit`	Maximum number of attributes that a MISP event will contain. It must be a non-negative integer (`0` means no limit). Section "Configuring Kaspersky Threat Feed App for MISP (on page 10)" contains recommendations on choosing the value of the `attributes_limit` parameter (by choosing the value of the `RECORDS_COUNT` parameter in the settings.py script). The `attributes_limit` parameter is optional. If it is not specified, no limit for the number of attributes is set.
`-nv` `--no_verification`	Disables the SSL certificate verification that is performed when connecting to a MISP instance by HTTPS. Use this parameter if you use a self-signed certificate on your MISP instance. Otherwise, the converter will lack the capability to add, modify, or delete events and attributes in MISP during work with the MISP API. This parameter is intended for evaluation purposes only. Using this parameter in a production environment may create security issues.

Kaspersky Threat Feed App for MISP features

If Kaspersky Threat Feed App for MISP is stopped during its work (for example, its process is not executed or the operating system is restarted), it resumes work after you run the master script run.py. If work does not resume, contact a Kaspersky representative. Alternatively, remove the imported events from the MISP instance, remove the contents of the `workdir` and `feed_util/feeds` directories, and remove the tool.pid file from the directory in which Kaspersky Threat Feed App for MISP is installed. Then run the master script run.py; the importing process will be performed from scratch.

The master script run.py and the importing script import_to_misp.py log their activities to stdout by default. We recommend that you save the log messages to a file (see section "Configuring Kaspersky Threat Feed App for MISP" on page 10) so that you can track the work performed by Kaspersky Threat Feed App for MISP.

Due to MISP restrictions, the MISP events that contain a large amount of attributes (more than 50 000) can be opened very slowly in the user interface or cannot be opened at all. This happens because MISP tries to get all attributes from a database and load the correlations between the attributes to memory. To get rid of these errors, we recommend that you increase the memory usage for the MISP instance. To do this, perform the following actions on the MISP computer:

1. Open the `/etc/php/7.1/apache2/php.ini` file for editing (the path may be different depending on the PHP version installed):

   ```
   vi /etc/php/7.1/apache2/php.ini
   ```

2. Specify the `memory_limit` parameter:

   ```
   memory_limit = 5G
   ```

3. Restart Apache:

   ```
   sudo systemctl restart apache2
   ```

kaspersky

AO Kaspersky Lab

Kaspersky is a world-renowned vendor of systems protecting computers against digital threats, including viruses and other malware, unsolicited email (spam), and network and hacking attacks.

In 2008, Kaspersky was rated among the world's top four leading vendors of information security software solutions for end users (IDC Worldwide Endpoint Security Revenue by Vendor). Kaspersky is the preferred vendor of computer protection systems for home users in Russia (IDC Endpoint Tracker 2014).

Kaspersky was founded in Russia in 1997. It has since grown into an international group of companies with 38 offices in 33 countries. The company employs more than 3,000 skilled professionals.

Products. Kaspersky products provide protection for all systems, from home computers to large corporate networks.

The personal product range includes security applications for desktop, laptop, and tablet computers, smartphones and other mobile devices.

The company offers protection and control solutions and technologies for workstations and mobile devices, virtual machines, file and web servers, mail gateways, and firewalls. The company's portfolio also features specialized products providing protection against DDoS attacks, protection for industrial control systems, and prevention of financial fraud. Used in conjunction with centralized management tools, these solutions ensure effective automated protection for companies and organizations of any size against computer threats. Kaspersky products are certified by major test laboratories, compatible with software from diverse vendors, and optimized to run on many hardware platforms.

Kaspersky virus analysts work around the clock. Every day they uncover hundreds of thousands of new computer threats, create tools to detect and disinfect them, and include their signatures in databases used by Kaspersky applications.

Technologies. Many technologies that are now part and parcel of modern anti-virus tools were originally developed by Kaspersky. It is no coincidence that many other developers use the Kaspersky Anti-Virus engine in their products, including: Alcatel-Lucent, Alt-N, Asus, BAE Systems, Blue Coat, Check Point, Cisco Meraki, Clearswift, D-Link, Facebook, General Dynamics, H3C, Juniper Networks, Lenovo, Microsoft, NETGEAR, Openwave Messaging, Parallels, Qualcomm, Samsung, Stormshield, Toshiba, Trustwave, Vertu, and ZyXEL. Many of the company's innovative technologies are patented.

Achievements. Over the years, Kaspersky has won hundreds of awards for its services in combating computer threats. Following tests and research conducted by the reputed Austrian test laboratory AV-Comparatives in 2014, Kaspersky ranked among the top two vendors by the number of Advanced+ certificates earned and was ultimately awarded the Top Rated certificate. But Kaspersky's main achievement is the loyalty of its users worldwide. The company's products and technologies protect more than 400 million users, and its corporate clients number more than 270,000.

Kaspersky website:	https://www.kaspersky.com
Virus encyclopedia:	https://securelist.com
Kaspersky VirusDesk:	https://virusdesk.kaspersky.com (for analyzing suspicious files and websites)
Kaspersky Community:	https://community.kaspersky.com

Trademark notices

Registered trademarks and service marks are the property of their respective owners.

Apache and the Apache feather logo are trademarks of The Apache Software Foundation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Python is a trademark or registered trademark of the Python Software Foundation.

Snort is a registered trademark or trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.