kaspersky

Kaspersky Threat Feed App for MISP

Product version: 2.1

Dear User,

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

Kaspersky reserves the right to amend this document without additional notification.

Kaspersky assumes no liability for the content, quality, relevance, or accuracy of any materials used in this document to which rights are held by third parties, or for any potential harms associated with use of the document.

Document revision date: 28.09.2020

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

About Kaspersky: (https://www.kaspersky.com/about/company)

Contents

About this document

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

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).

About Kaspersky Threat Feed App for MISP

Every record in Kaspersky Threat Data Feeds contains the following information:

- Indicators to match against your events and logs.
- Actionable context to provide actionable intelligence for indicators.

Indicators and context fields from the feeds are described in Supported feeds (on page 11).

Kaspersky Threat Feed App for MISP does the following:

- Downloads a new version of selected Kaspersky Threat Data Feeds.
- Converts new records from these feeds to MISP format.
- Removes obsolete records from the MISP instance by using MISP API.

To add converted feeds to MISP, you must add them as custom feeds from the MISP web interface. Every record from Kaspersky Threat Data Feeds is imported as a MISP event.

In this chapter

Distribution kit

The Kaspersky Threat Feed App for MISP distribution kit contains the following files and directories.

Table 1. Package contents

Item	Description
base_processes.py	Python module necessary for the operation of converting feeds.
defs.py	Python module containing settings and definitions (see Installing Kaspersky Threat Feed App for MISP (on page 13)).
feed_util.py	Python module necessary for the operation of Feed Utility.
kl_to_misp_converter.py	Python module necessary for the operation of converting feeds.
main.py	Feed converter.
misp_api.py	Python module for the importing utility.
requirements.txt	Lists the Python packages necessary for the operation of the converter.
settings.py	File containing settings (see Configuring Kaspersky Threat Feed App for MISP (on page 15))
utils.py	Python module necessary for the operation of converting feeds.
doc/EULA.txt	End User License Agreement (EULA).
doc/Kaspersky Threat Feed App for MISP.pdf	This documentation.
doc/legal_notices.txt	Text file with legal notices for the converter and Feed Utility. This file contains information about third-party code used by Kaspersky Threat Feed App for MISP.
feed_util/feeds.pem	Certificate needed for getting access to demo feeds.
feed_util/kl_feed_util	Feed Utility binary file. Feed Utility downloads Kaspersky Threat Data Feeds.
feed_util/kl_feed_compiler	Binary file used by Feed Utility to compile feeds.
feed_util/template.conf	File used by the main.py script as a Feed Utility configuration file template.
feed_util/libssp.so.0	Auxiliary library.

System requirements

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

Supported operating systems

The converter can be run on the following operating systems:

- Linux® x64

Software requirements

The converter requires Python 3.6 or later. The Python packages listed in requirements.txt must also be installed.

Disk space requirements

Every feed from Kaspersky Threat Data Feeds requires from 20 megabytes (MB) to 3.5 gigabytes (GB) of hard disk space, depending on the feed size.

If all feeds (see section "Supported feeds" on page 11) are converted, it is recommended to have at least 10 GB of hard disk space available. This disk space is used during the feed conversion process.

Make sure that enough index nodes (inodes) are available on the disk where Kaspersky Threat Data Feeds will be converted. The inodes count is important because the inodes limit is close to the file count limit. If all available feeds are enabled, as a result of MISP conversion, about 2.5M to 3M small JSON files representing MISP events will be created. If the inode count is not large enough, the conversion process can be interrupted with an insufficient disk space error. The error in this case is not caused by insufficient disk space but by the insufficient count of inodes.

The recommended amount of inodes is 3M. You can check the available amount with the `df -ih` command.

If the free inodes count is lower than 3M, it is recommended to add a new filesystem with a lower inodes to blocks count ratio specified in its `usage-type`. For example, for ext4 file system the value should be one inode for every 4096 bytes. For mode information, see https://wiki.archlinux.org/index.php/Ext4#Bytes-per-inode_ratio.

RAM requirements

Kaspersky Threat Feed App for MISP uses a maximum of 500 MB of RAM.

CPU requirements

Kaspersky Threat Feed App for MISP requires a multi-core CPU with two or more cores.

General workflow

Kaspersky Threat Feed App for MISP works as follows:

1. Feed Utility is used to download feeds.

   Feed Utility is a tool that downloads and filters Kaspersky Threat Data Feeds according to rules defined in its configuration file. For more information, see https://click.kaspersky.com/?hl=en-US&link=online_help&pid=CyberTrace&version=1.0&helpid=171402.
2. For imports other than the first import, the diff of a feed is created.
3. The new records of the feed are converted to MISP-format files and saved to the specified directory.
4. Obsolete records are removed from the MISP instance using its REST API.

Kaspersky Threat Feed App for MISP can create two kinds of updates:

- A full diff update

  This update contains records with changed context fields. It take a significant amount of time. The converter performs a full diff update only at intervals (in hours) defined by the `full_update_interval_h` parameter (see section "Command-line parameters" on page 17).
- An update with a truncated diff

This update only adds records with new indicators of compromise (IOC) and deletes records with obsolete IOC. The converter performs this update at intervals defined by the schedule specified for the `cron` utility (see section "Scheduling feeds conversion" on page 19).

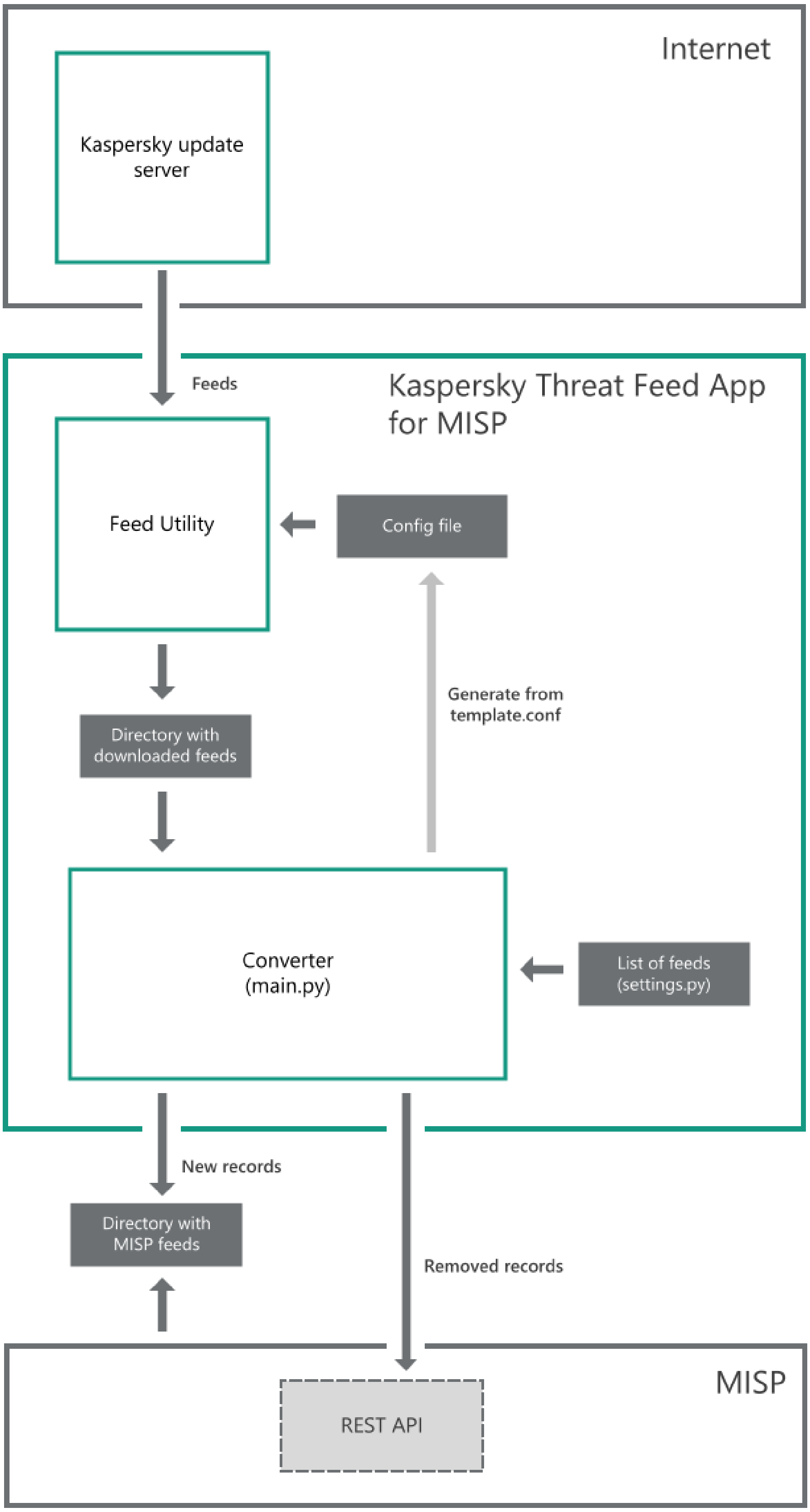
Internet
Kaspersky update server
Feeds
Kaspersky Threat Feed App for MISP
Feed Utility
Config file
Directory with downloaded feeds
Generate from template.conf
Converter (main.py)
List of feeds (settings.py)
New records
Directory with MISP feeds
Removed records
MISP
REST API

Figure 1: Kaspersky Threat Feed App for MISP workflow

Supported feeds

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

Commercial feeds

Commercial feeds are regular Kaspersky Threat Data Feeds.

The following commercial feeds are available:

- IP Reputation Data Feed

  A set of IP addresses with context that cover different categories of suspicious and malicious hosts.

- Malicious Hash Data Feed

  A set of file hashes with context that cover the most dangerous, prevalent, or emerging malware.

- Mobile Malicious Hash Data Feed

  A set of file hashes with 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 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.

- Mobile Botnet Data Feed

  A set of URLs with context that cover mobile botnet C&C servers.

- Ransomware URL Data Feed

  A set of URLs, domains, and hosts with context that cover ransomware links and websites.

- Malicious URL Exact Data Feed

  A set of exact URLs with context that cover malicious websites and web pages.

- Botnet CnC URL Exact Data Feed

  A set of exact URLs with context that cover desktop botnet C&C servers and related malicious objects.

- Phishing URL Exact Data Feed

  A set of exact URLs with context that cover phishing websites and web pages.

- APT IP Data Feed

  A set of IP addresses that belong to the infrastructure used in APT campaigns.

- APT URL Data Feed

  A set of domains that belong to the infrastructure used in APT campaigns.

- APT Hash Data Feed

  A set of hashes that cover malicious artifacts used by APT actors to conduct APT campaigns.

- Vulnerability Data Feed

  A set of file hashes with context that cover vulnerabilities in applications and cover exploits that use those vulnerabilities.

Demo feeds

Demo feeds can be used for evaluation purposes. These feeds provide lower detection rates in comparison with their corresponding commercial versions.

The following demo feeds are available:

- Demo Botnet CnC URL Data Feed

  Provides lower detection rates in comparison with Botnet CnC URL Data Feed.
- Demo IP Reputation Data Feed

  Provides lower detection rates in comparison with IP Reputation Data Feed.
- Demo Malicious Hash Data Feed

  Provides lower detection rates in comparison with Malicious Hash Data Feed.

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.

Installing Kaspersky Threat Feed App for MISP

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

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

1. Unpack the distribution kit to the desired directory.

   This directory is called `%service_dir%` in this document.

2. Rename your certificate for downloading feeds to `feeds.pem` and copy it to the `%service_dir%/feed_util` subdirectory.

3. Read the End User License Agreement (EULA). You can find the terms of the EULA in the `%service_dir%/EULA.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. Open the `%service_dir%/feed_util/template.conf` file for editing.

5. 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.

6. Save and close the template.conf file.

7. By default, the converted MISP-format feeds are saved to the `%service_dir%/workdir` directory. If you want to save them to a different directory, do the following:

   a. Open the `%service_dir%/defs.py` file for editing.

   b. Find the following line:

      ```
      WORK_DIR = os.path.join(BASE_DIR, 'workdir')
      ```

   c. In the `WORK_DIR` parameter, specify the path to the directory where you want to store MISP-format feeds.

      You can set `WORK_DIR` to an absolute path as follows:

      `WORK_DIR = '%absolute_path%'` (replace `%absolute_path%` with an absolute path to the directory).

   d. Save and close the `%service_dir%/defs.py` file.

   > When the converter and the MISP instance operate on the same computer, the user account that runs the MISP instance must have access rights to the `WORK_DIR` directory so that the MISP instance can download the converted feeds.

8. 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 %service_dir%/requirements.txt
   ```

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

9. Configure Kaspersky Threat Feed App (see section "Configuring Kaspersky Threat Feed App for MISP" on page 15) for MISP by editing the `%service_dir%/settings.py` file.
10. 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.
11. Run the first feed conversion process manually. For more information about running the converter from the command line, see Command-line parameters (on page 17).

After the installation, you can schedule the feeds conversion (see section "Scheduling feeds conversion" on page 19) by using the `cron` utility.

Configuring Kaspersky Threat Feed App for MISP

Settings for Kaspersky Threat Feed App for MISP are specified in the settings.py file.

settings.py

The settings.py file contains the following parameters:

- `RECORDS_COUNT`

  Defines the maximum number of records imported from Kaspersky Threat Data Feeds.

  It is not recommended to change this value.

  If this value is `0`, all records are imported.

  The default value for this parameter is `200000`.

- `FEEDS`

  A dictionary that contains the identifiers and names of feeds. Also, the dictionary can contain the filters applied to the feed records that will be re-used by Feed Utility without modification. Uncomment those feeds that must be converted to MISP format. Do not change feed names or identifiers.

  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 (that is, the `threat_score` field for which is equal to or exceeds 75).

- `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 compared to 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 `%service_dir%/feed_util/template.conf` file. For more information about Feed Utility logging, see Configuration file parameters (Feed Utility) in the online documentation for Kaspersky CyberTrace.

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 `%service_dir%` directory. Make sure that a user that runs the converter has sufficient rights to write to this file.

- `PROCESS_TIMEOUT`

  Internal parameter.

  It is not recommended to change this value.

  The default value for this parameter is `2`.

- `QUEUE_SIZE`

  Internal parameter.

  It is not recommended to change this value.

  The default value for this parameter is `10000`.

- `WORKERS_COUNT`

  The number of processes that are created when feeds are processed.

  The recommended number of processes is `(CPU_CORES * 2) - 2`, where `CPU_CORES` is the number of CPU cores.

  If the target computer has two cores, set this value to `1`.

- `VULN_FILES_COUNT`

  Defines the maximum number of vulnerable files added to a MISP event from a feed.

  If this value is `0`, no vulnerable files are added.

  You can set this parameter only for Vulnerability Data Feed.

- `EXPLOITS_FILES_COUNT`

  Defines the maximum number of exploits added to a MISP event from a feed.

  If this value is `0`, no exploits are added.

  You can set this parameter only for Vulnerability Data Feed.

- `ADD_HAS_EXPLOITS_FIELD`

  Flag indicating that the `has_exploits` field is added to a MISP event. This field takes the `1` value, if a feed record for vulnerable files contains exploits. Otherwise, if a feed record does not contain exploits, the value is `0`.

  If the value of `ADD_HAS_EXPLOITS_FIELD` is `True`, the `has_exploits` field is added.

  You can set this parameter only for Vulnerability Data Feed.

Using Kaspersky Threat Feed App for MISP

This chapter explains how to use Kaspersky Threat Feed App for MISP.

Command-line parameters

The converter is a console application. You can invoke it from the command line.

Syntax

The converter uses the following syntax:

```
python main.py --misp_url <MISP_URL> --auth_key
<AUTH_key_of_MISP_instance> [--full_update_interval_h
<full_update_interval_hours>] [-nv] [-h]
```

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

Options

The following options are available:

- `--misp_url <MISP_URL>`

  URL or IP address of a MISP instance.

  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 delete obsolete events from MISP.

- `--auth_key <AUTH_KEY_MISP_INSTANCE>`

  AUTH key of a MISP instance.

  The AUTH key is available in the MISP UI interface.

- `--full_update_interval_h <full_update_interval_hours>`

  Interval (in hours) between full updates of Kaspersky Threat Data Feeds in MISP. For more information, see General workflow (on page 8).

  By default, this parameter is `12`.

- `-nv, --no_verification`

  Disables the SSL certificate verification 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 not be able to delete obsolete events from MISP.

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

- `-h, --help`

  Prints a help message to the console and exits.

Loading converted feeds to MISP

When the converter finishes processing the feeds, it creates a set of folders in the `WORK_DIR` folder that is defined during the installation (see section "Installing Kaspersky Threat Feed App for MISP" on page 13). Each folder represents one feed from Kaspersky Threat Data Feeds as a MISP Feed.

To load those feeds into a MISP instance, add a set of MISP feeds to your MISP instance according to the MISP documentation.

After the feeds are added to a MISP instance, you can configure the fetching of events from those feeds using the MISP UI. As an alternative, you can use MISP API. For example, you can use the `curl` utility:

```
curl --insecure -i -X GET -H "Authorization: %auth_key%" -H "Accept:
application/json" -H "content-type: application/json"
%misp_url%/feeds/fetchFromFeed/%feed_id%
```

In the above command, replace:

- `%auth_key%` with the AUTH key of a MISP instance (see section "Command-line parameters" on page 17),
- `%misp_url%` with the URL or IP address of a MISP instance (see section "Command-line parameters" on page 17), and
- `%feed_id%` with the feed identifier that was assigned to it by MISP. Feed identifiers are available in the MISP UI.

The `--insecure` parameter causes `curl` to establish insecure SSL connections. This may create security issues. Use it only for evaluation purposes.

The fetching of events from feeds was tested on a computer with a 3.0 GHz CPU, four cores, and 24 GB of RAM. Further improving the hardware of the computer does not significantly affect the performance. For a feed that includes no more than 200 000 records, the first fetching process takes the following approximate amounts of time:

- IP Reputation Data Feed—20 hours
- Malicious Hash Data Feed—40 hours
- Mobile Malicious Hash Data Feed—40 hours
- P-SMS Trojan Data Feed—6 hours 30 minutes
- Mobile Botnet Data Feed—15 hours
- Ransomware URL Data Feed—40 minutes

- Malicious URL Exact Data Feed—15 hours
- Botnet CnC URL Exact Data Feed—40 hours
- Phishing URL Exact Data Feed—40 hours
- APT Hash Data Feed—2 hours 30 minutes
- APT IP Data Feed—30 minutes
- APT URL Data Feed—1 hour 30 minutes
- Vulnerability Data Feed—3 hours

Scheduling feeds conversion

After you make the first conversion and import feeds into a MISP instance, you can make Kaspersky Threat Feed App for MISP run periodically at a specific interval using the `cron` utility.

Update interval

The specific update interval of a feed depends on a feed type and on its limit of records.

To schedule the conversion to be done every 30 minutes, run the following commands from the command line:

```
crontab -l > /tmp/misp_feeds_conv_crontab
echo "*/30 * * * * python %service_dir%/main.py" >>
/tmp/misp_feeds_conv_crontab
crontab /tmp/misp_feeds_conv_crontab && echo "Success" || echo "Failed"
```

The `cron` utility will now execute the main.py script every 30 minutes.

If the converter script does not finish the feed conversion process by the time the `cron` utility runs the converter again, then an attempt to run it again will produce an error. After the converter script finishes the feed conversion process, it can be run again.

Conversion performance

The initial conversion process converts all records (within `RECORDS_COUNT`) from all the enabled feeds to MISP format. Subsequent converter calls make a diff with existing feeds but make no changes in records that were changed. The full update will launch only if the interval between the time of the last full update and the current moment exceeds the value defined in the `full_update_interval_h` parameter (by default, 12 hours).

Loading all the feeds into MISP can take up to several days, but you can create a pool of MISP instances with one instance for each feed. In this case, set up the converter on each node to convert only one feed.

If done regularly, subsequent feed updates should not take more than an hour. The less frequently the updates are done, the bigger the diff and, consequently, the longer the update process.

Restoring the converter operation after a failure

If the main.py script (the converter utility) 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 main.py script. If work starts with errors, contact your technical account manager (TAM).

Alternatively, remove the imported events from the MISP instance, remove the contents of the `WORK_DIR` and `feed_util/feeds` directories, and remove the `tool.pid` file from the `WORK_DIR` directory. Then, run the main.py script; the converting process will be performed from the beginning.

Troubleshooting

Problems may arise during the conversion of a JSON-formatted feed to MISP format.

The main.py script of Kaspersky Threat Feed App for MISP prints "Tool failed" or returns an exception

To solve this problem, try the following actions:

- Accept or make sure that you have accepted the EULA in the `%service_dir%/feed_util/template.conf` file. For information on how to accept the EULA, see Installing Kaspersky Threat Feed App for MISP (on page 13).
- Make sure that the feeds.pem certificate is located in the `%service_dir%/feed_util` directory. If the file is not is this directory, copy it there.
- Install or make sure that you have installed the libraries listed in the requirements.txt file (see section "Installing Kaspersky Threat Feed App for MISP" on page 13).

Trademark notices

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

Apple, iPhone are trademarks of Apple Inc., registered in the U.S. and other countries.

Google, Android are trademarks of Google, Inc.

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.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

Information about third-party code

Information about third-party code is contained in a `doc/legal_notices.txt` file in the Kaspersky Threat Feed App for MISP distribution kit.

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