ThreatQuotient

MISP Export Connector Guide

Version 1.1.0

July 20, 2021

ThreatQuotient
11400 Commerce Park Dr., Suite 200
Reston, VA 20191

Support
Email: support@threatq.com
Web: support.threatq.com
Phone: 703.574.9893

Contents

Warning and Disclaimer

ThreatQuotient, Inc. provides this document “as is”, without representation or warranty of any kind, express or implied, including without limitation any warranty concerning the accuracy, adequacy, or completeness of such information contained herein. ThreatQuotient, Inc. does not assume responsibility for the use or inability to use the software product as a result of providing this information.

Versioning

- Current integration version: `1.1.0`
- Supported on ThreatQ versions >= `4.30.0`

There are two versions of this integration:

- Python 2 version
- Python 3 version

Introduction

The MISP Export connector is a uni-directional connector which will create a MISP Event and store indicators of compromise (IoCs) in the MISP Event for the current day.

There is a limitation in the number of attributes a MISP Event can hold. If there are more than 25,000 IoCs to be exported on a given day, the code will create a new MISP Event for each 25,000 IoC collection. This connector requires interaction within the ThreatQ UI as well as at the command line.

The MISP server's php.ini file must be modified in order to store 25,000 indicators. See the PHP Configuration section below for more information.

Prerequisites

PHP Configuration

The following are the minimum settings required in order for MISP to store ThreatQ indicators:

```
max_execution_time = 300
memory_limit = 512M
upload_max_filesize = 50M
post_max_size = 50M
```

Installation

The connector can be installed from the ThreatQuotient repository with YUM credentials or offline via a .whl file.

> **Upgrading Users** - Review the Change Log for updates to configuration parameters before updating. If there are changes to the configuration file (new/removed parameters), you must first delete the previous version's configuration file before proceeding with the install steps listed below. Failure to delete the previous configuration file will result in the connector failing.

1. Install the connector using one of the following methods:

   ThreatQ Repository

   a. Run the following command:

   ```
   <> pip install tq_conn_misp_export
   ```

   Offline via .whl file

   To install this connector from a wheel file, the wheel file (.whl) will need to be copied via SCP into your ThreatQ instance.

   a. Download the connector whl file with its dependencies:

   ```
   <> mkdir /tmp/ tq_conn_misp_export

      pip download tq_conn_misp_export -d

      /tmp/ tq_conn_misp_export/
   ```

   b. Archive the folder with the .whl files:

   ```
   <> tar -czvf tq_conn_misp_export.tgz /tmp/
      tq_conn_misp_export/
   ```

   c. Transfer all the whl files, the connector and all the dependencies, to the ThreatQ instance.

   d. Open the archive on ThreatQ:

   ```
   <> tar -xvf tq_conn_misp_export.tgz
   ```

e. Install the connector on the ThreatQ instance.

> The example assumes that all the whl files are copied to `/tmp/conn` on the ThreatQ instance.

```
<> pip install /tmp/conn/ tq_conn_misp_export-<version>-
   <python version>-none-any.whl --no-index --find-links /
   tmp/conn/
```

> A driver called `tq-conn-misp-export` will be installed. After installing with `pip` or `setup.py`, a script stub will appear in `/usr/bin/tq-conn-misp-export`.

2. Once the application has been installed, a directory structure must be created for all configuration, logs and files, using the `mkdir -p` command. Use the commands below to create the required directories:

```
<> mkdir -p /etc/tq_labs/
   mkdir -p /var/log/tq_labs
```

3. Perform an initial run using the following command:

```
<> tq-conn-misp-export -v3 -ll /var/log/tq_labs/ -c /etc/tq_labs/
```

4. Enter the following parameters when prompted:

PARAMETER	DESCRIPTION
ThreatQ Host	This is the host of the ThreatQ instance, either the IP Address or Hostname as resolvable by ThreatQ.
Client ID	This is the OAuth id that can be found at Settings Gear → User Management → API details within the user's details.
Email Address	This is the User in the ThreatQ System for integrations.
Password	The password for the above ThreatQ account.

PARAMETER	DESCRIPTION
Status	This is the default status for objects that are created by this Integration.

Example Output

```
tq-conn-misp-export -v3 -ll /var/log/tq_labs/ -c /etc/tq_labs/
ThreatQ Host: <ThreatQ Host IP or Hostname>
Client ID: <ClientID>
E-Mail Address: <EMAIL ADDRESS>
Password: <PASSWORD>
Status: Review
Connector configured. Set information in UI
```

You will still need to configure and then enable the connector.

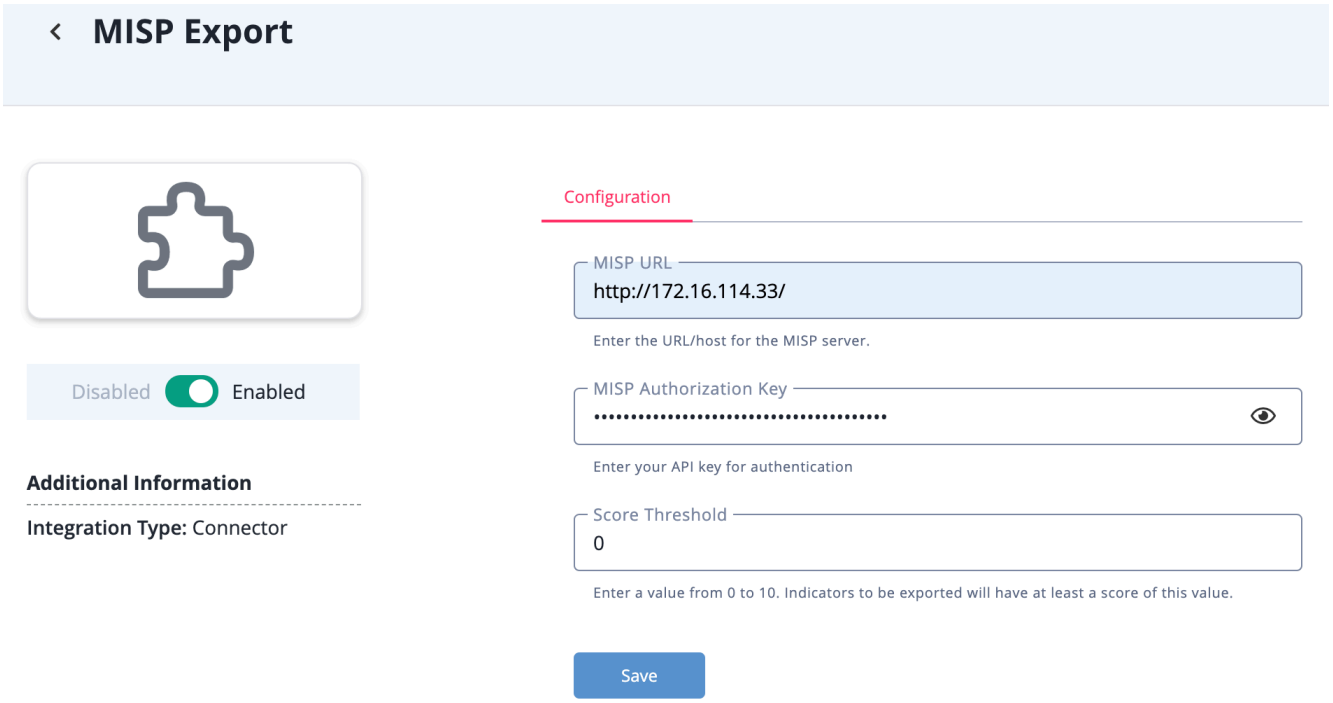
Configuration

> ThreatQuotient does not issue API keys for third-party vendors. Contact the specific vendor to obtain API keys and other integration-related credentials.

To configure the integration:

1. Navigate to your integrations management page in ThreatQ.
2. Select the **Labs** option from the *Category* dropdown (optional).
3. Click on the integration to open its details page.
4. Enter the following parameters under the **Configuration** tab:

PARAMETER	DESCRIPTION
MISP Host URL	The URL for the applicable MISP SIEM.
MISP Authorization Key	Provides authorization into the MISP API. The value can be found on the MISP server under **Administration/List Users/Authkey**.
Score Threshold	Active indicators to be exported will have at least a score of the value set in this field. The default setting is 0.



5. Review any additional settings available, make any changes if needed, and click on **Save**.
6. Click on the toggle switch, located above the *Additional Information* section, to enable it.

Usage

Use the following command to execute the driver:

```
<> tq-conn-misp-export -v3 -ll /var/log/tq_labs/ -c /etc/tq_labs/
```

Command Line Arguments

This connector supports the following custom command line arguments:

ARGUMENT	DESCRIPTION
`-h`, `--help`	Shows this help message and exits.
`-n NAME, --name NAME`	This sets the name for this connector. In some cases, it is useful to have multiple connectors of the same type executing against a single TQ instance. For example, the Syslog Exporter can be run against multiple target and multiple exports, each with their own name and configuration
`-d, --no-differential`	If exports are used in this connector, this will turn 'off' the differential flag for the execution. This allows debugging and testing to be done on export endpoints without having to rebuild the exports after the test. ⚠ This should never be used in production.
`-ll LOGLOCATION, --loglocation LOGLOCATION`	Sets the logging location for the connector. The location should exist and be writable by the current. A special value of 'stdout' means to log to the console (this happens by default).
`-c CONFIG, --config CONFIG`	This is the location of the configuration file for the connector. This location must be readable and writable by the current user. If no config file path is given, the current directory will be used. This file is

ARGUMENT	DESCRIPTION
	also where some information from each run of the connector may be put (last run time, private oauth, etc.)
`-v {1,2,3}`, `--verbosity {1,2,3}`	This is the logging verbosity level where **3** means everything. The default setting is **1** (Warning).
`-ep, --external-proxy`	This allows you to use the proxy that is specified in the ThreatQ UI.  This specifies an internet facing proxy, NOT a proxy to the TQ instance.
`-ds, --disable-ssl`	Adding this flag will disable SSL verification when contacting the MISP API.
`-md, --mispdebug`	Enable debug mode in MISP API.

CRON

Automatic CRON configuration has been removed from this script. To run this script on a recurring basis, use CRON or some other jobs scheduler. The argument in the CRON script must specify the config and log locations.

Add an entry to your Linux crontab to execute the connector at a recurring interval. Depending on how quickly you need updates, this can be run multiple times a day (no more than once an hour) or a few times a week.

In the example below, the command will execute the connector every two hours.

1. Log into your ThreatQ host via a CLI terminal session.
2. Enter the following command:

   ```
   <> crontab -e
   ```

   This will enable the editing of the crontab, using vi. Depending on how often you wish the cronjob to run, you will need to adjust the time to suit the environment.

3. Enter the commands below:

   Every 2 Hours Example

   ```
   <> 0 */2 * * * tq-conn-mips-export -c /etc/tq_labs/ -ll /var/log/
      tq_labs/ -v3
   ```

4. Save and exit CRON.

Change Log

- **Version 1.1.0**
  - Added Python 3 support.
- **Version 1.0.7**
  - Updated connector to use the ThreatQ Threat Library opposed to the legacy indicator search.
- **Version 1.0.6**
  - Improved support for Python 2.
  - Updated tqdriver.
- **Version 1.0.5**
  - Removed prompt for default status. Status ID is now set to a hard-coded value.
- **Version 1.0.4**
  - Fixed an issue where Score Threshold excluded scores of 10.
- **Version 1.0.3**
  - Added flags to diable SSL verify and enable MISP debug mode.
  - Fixed a bug with the Score Threshold configuration option.
- **Version 1.0.2**
  - Added proxy support.
- **Version 1.0.1**
  - Fixed a scoring storage issue.
- **Version 1.0.0**
  - Initial Release