ThreatQuotient

FireEye CMS Connector Guide

Version 3.3.1

April 06, 2021

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

Support

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

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.

Contents

Versioning

- Current integration version: 3.3.1
- Supported on ThreatQ versions >= 4.34.0

There are two versions of this integration:

- Python 2 version
- Python 3 version

Introduction

This is a custom connector for a FireEye CMS instance. This connector is meant to attach to a single FireEye CMS instance.

The connector has several purposes:

- Pull alerts from FireEye CMS and upload the data as indicators and events to a ThreatQ instance. The events are tagged as "Malware" type events.
- Upload indicators from ThreatQ to the FireEye CMS instance by inputting one or multiple Data Collections in the UI.

> The `-n` flag may be used for multiple instances, but it is not a tested use case.

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_fireeye_cms
   ```

   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_fireeye_cms
   pip download tq_conn_fireeye_cms -d /tmp/tq_conn_fireeye_cms/
   ```

   b. Archive the folder with the .whl files:

   ```
   tar -czvf tq_conn_fireeye_cms.tgz /tmp/tq_conn_fireeye_cms/
   ```

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

```
pip install /tmp/conn/tq_conn_fireeye_cms-3.3.0-py2-none-any.whl --no-index --find-links /tmp/conn/
```

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

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-fireeye-cms -c /path/to/config/directory/ -ll /path/to/log/directory/ -ds -v3
```

> The -ds flag is used to disable SSL verification. This should be used if the FireEye CMS instance this is connected to uses a self-signed certificate. By default, FireEye CMS uses a self-signed certificate.
>
> To use the proxy settings specified in your ThreatQ UI, append the -ep flag to the end.
>
> If you want to parse data from a JSON file (instead of the API), append the -f /path/to/your/file flag to the end of your command. This will parse the file, but not change the time internal time record for when this connector last connected to FireEye CMS. Ingesting duplicate data will not cause issues, as the data will be recorded the same way.
>
> See the Command Line Arguments section for more details.

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.
Status	This is the default status for objects that are created by this Integration. It is common to set this to "Review", but Organization SOPs should be respected when setting this.

Example Output

```
tq-conn-fireeye-cms -c /opt/tq-integrations/fireeye_cms/config/ -ll /opt/tq-integrations/fireeye_cms/logs/ -ds -v3
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).

   > If you are installing the connector for the first time, it will be located under the **Disabled** tab.

3. Click on the integration to open its details page.
4. Enter the following parameter under the **Configuration** tab:

PARAMETER	DESCRIPTION
Host	The hostname or the IP Address of the FireEye CMS instance.
Username	The FireEye CMS instance username.
Password	The password associated with the username above.
Alert Duration	(The interval in which to search for alerts. Options include (hours): 1, 2, 6, 12, 24, and 48.
Data Collection	The Data Collection names in ThreatQ that will used to import custom IoCs to FireEye.

5. Review the **Settings** configuration, 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

The following section contains steps, commands, and arguments required to run the connector.

Executing the FireEye CMS Driver

Use the following command to execute the FireEye CMS driver:

```
tq-conn-fireeye-cms -c /path/to/config/directory/ -ll /path/to/log/directory/ -ds -v3
```

Executing the FireEye IoC Driver

Perform the following steps to run the FireEye IoC Driver:

1. Log into the ThreatQ platform and locate the FireEye CMS installation under My Integrations.
2. Confirm that Data Collections / Saved Searches field has at least one search added to it.

   > If including multiple data collections / saved searches, use a comma separated format as demonstrated in the example image below

3. Run the following command:

   ```
   tq-conn-fireeye-ioc-sync -c /path/to/config/directory/ -ll /path/to/log/directory/ -ds -v 3
   ```

Command Line Arguments

This connector supports the following custom command line arguments:

ARGUMENT	DESCRIPTION
-h, --help	Shows this help message and exits.
-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 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).
-f, --file	Allows you to specify parsing JSON of a file instead of using the API endpoint of FireEye CMS. Used for offline data pulls.
-ds, --disable_ssl	This allows you to disable SSL verification to all requests to the FireEye CMS API.
-ep, -external-proxy	This allows you to use the proxy that is specified in the ThreatQ UI.

Most commonly, if you want to see the output in the command line, you will execute:

```
tq-conn-fireeye-cms -ll stdout -v3
```

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-fireeye-cms -c /path/to/config/directory/ -ll /path/to/log/
   directory/ -ll /path/to/config/directory/ -v3
   ```

4. Save and exit CRON.

Change Log

- **Version 3.3.1**
    - Updated Parameter naming - **Saved Search** is now known as **Data Collection**.
- **Version 3.3.0**
    - Added Python 3 support.
    - Updated search system to use Data Collections / Saved Searches.
- **Version 3.2.1**
    - Added the ability to retrieve and upload data from FireEye CMS.
- **Version 2.0.0**
    - Minor documentation updates/enhancements.
- **Version 1.1.0**
    - Minor documentation updates/enhancements.
- **Version 1.0.0**
    - Initial Release