ThreatQuotient

FireEye EX Connector Guide

Version 1.2.1

April 03, 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: 1.2.1
- Supported on ThreatQ versions >= 4.34.0

There are two versions of this integration:

- Python 2 version
- Python 3 version

Introduction

The FireEye EX custom connector is a bi-directional integration that is used to:

1. Upload YARA rules from a data collection in ThreatQ to FireEye
2. Search for alerts in FireEye EX and create events from those alerts in ThreatQ, including any indicators that have been found in the alerts.

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_ex
   ```

   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_ex
   pip download tq_conn_fireeye_ex -d /tmp/tq_conn_fireeye_ex/
   ```

   b. Archive the folder with the .whl files:

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

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

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

2. Once the application has been installed, a directory structure must be created for all configuration, logs and files, using the mkdir -p command. See example below:

Creating Integration Directories Example

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

3. Perform an initial run using the following command:

```
tq-conn-fireeye-ex -c /path/to/config/directory/ -ll /path/to/log/directory/ -v VERBOSITY_LEVEL
```

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

Example Output

```
tq-conn-fireeye-ex -c /path/to/config/directory/ -ll /path/to/log/directory/ -v VERBOSITY_LEVEL
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
IP/Hostname	The hostname or the IP Address of the FireEye EX.
Port	Port for communicating with the FireEye EX instance. Default: 443.
Username	The username for connecting to FireEye.
Password	The password associated with the username above.
Data Collection with YARA Rules to send to FireEye EX	The name of the data collection in the ThreatQ instance with YARA rules to send to FireEye EX.
Get Alerts from FireEye EX	Optional - Check the provided checkbox if you want to get alerts from FireEye EX and create them as events in ThreatQ.

PARAMETER	DESCRIPTION
Historical Time Frame in Hours	Enter the historical number of hours to search for alerts. The maximum historical timeframe is 12 hours.

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

Use the following command to execute the driver:

```
tq-conn-fireeye-ex -c /path/to/config/directory/ -ll /path/to/log/directory/ -v3
VERBOSITY_LEVEL
```

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. The default setting is **1** (Warning).
-ds, --disable_ssl	This allows you to disable SSL verification to all requests when contacting the third-party API.
-ep, -external-proxy	This allows you to use the proxy that is specified in the ThreatQ UI.

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-ex -c /path/to/config/directory/ -ll /path/to/log/directory/ -ll /path/to/config/directory/ -v3
   ```

4. Save and exit CRON.

Change Log

- **Version 1.2.1**
  - Updated connector to use ThreatQ Threat Library Data Collections opposed to Saved Searches.
  - Improved logging capability.
  - Updated connector to reflect changes to the ThreatQ SDK.
- **Version 1.2.0**
  - Added Python 3 support.
- **Version 1.1.0**
  - YARA Rules are now pushed to FireEye EX host/ip opposed to FireEye EX.
  - Updated the proxy method for request session.
- **Version 1.0.0**
  - Initial Release