ThreatQuotient

ThreatQ Keyword Alerts Connector Guide

Version 1.0.0 rev-b

March 23, 2022

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

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.

Support

This integration is designated as **ThreatQ Supported**.

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

Integrations/apps/add-ons designated as **ThreatQ Supported** are fully supported by ThreatQuotient's Customer Support team.

ThreatQuotient strives to ensure all ThreatQ Supported integrations will work with the current version of ThreatQuotient software at the time of initial publishing. This applies for both Hosted instance and Non-Hosted instance customers.

> ⚠️ ThreatQuotient does not provide support or maintenance for integrations, apps, or add-ons published by any party other than ThreatQuotient, including third-party developers.

Versioning

5

- Current integration version: `1.0.0`
- Compatible with ThreatQ versions >= `4.30.0`
- Python version = 3.6

Introduction

The ThreatQ Keyword Alerts Connector for ThreatQuotient enables users to be alerted via email when a set of keywords are matched on incoming threat intelligence. Users can be alerted on new and updated objects such as adversaries, malware, and reports, that reference a keyword.

Prerequisites

Review the following requirements before attempting to install the connector.

Time Zone

You should ensure all ThreatQ devices are set to the correct time, time zone, and date (UTC is recommended), and using a clock source available to all.

To identify which time zone is closest to your present location, use the `timedatectl` command with the list-`timezones` command line option.

For example, enter the following command to list all available time zones in Europe:

```
timedatectl list-timezones | grep Europe
Europe/Amsterdam
Europe/Athens
Europe/Belgrade
Europe/Berlin
```

Enter the following command, as root, to change the time zone to UTC:

```
timedatectl set-timezone UTC
```

PIP.conf

Prior to ThreatQ version 4.10, you were required to modify your system's pip.conf to use the ThreatQ integrations python repo, also known as DevPi. This functionality was made available upon an initial install of 4.10. If you have upgraded to 4.10 from a previous version, you will need to modify the pip.conf on your environment to the following (replacing username and password with your information).

```
[global]
   index-url = https://system-updates.threatq.com/pypi
   extra-index-url = https://<username>:<password>@extensions.threatq.com/threatq/integrations
                     https://<username>:<password>@extensions.threatq.com/threatq/sdk
```

Integration Dependencies

> ⚠️ The integration must be installed in a python 3.6 environment.

The following is a list of required dependencies for the integration. These dependencies are downloaded and installed during the installation process. If you are an Air Gapped Data Sync (AGDS) user, or run an instance that cannot connect to network services outside of your infrastructure, you will need to download and install these dependencies separately as the integration will not be able to download them during the install process.

> 📝 Items listed in bold are pinned to a specific version. In these cases, you should download the version specified to ensure proper function of the integration.

DEPENDENCY	VERSION	NOTES
threatqsdk	>= 1.8.4	N/A
threatqcc	>= 1.4.1	N/A
python-dateutil	>= 2.8.2	N/A

Installation

The following provides you with steps on installing a Python 3 Virtual Environment and installing the connector.

Creating a Python 3.6 Virtual Environment

Run the following commands to create the virtual environment:

```
mkdir /opt/tqvenv/
sudo yum install -y python36 python36-libs python36-devel python36-pip
python3.6 -m venv /opt/tqvenv/<environment_name>
source /opt/tqvenv/<environment_name>/bin/activate
pip install --upgrade pip
pip install setuptools==59.6.0
pip install threatqsdk threatqcc
pip install python-dateutil
```

Proceed to installing the connector.

Installing the Connector

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. Activate the virtual environment:

    ```
    source /opt/tqvenv/<environment_name>/bin/activate
    ```

    b. Run the following command:

    ```
    pip install tq_conn_keyword_alerts
    ```

    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 from a separate device with internet access:

    ```
    mkdir /tmp/tq_conn_keyword_alerts

    pip download tq_conn_keyword_alerts -d
    /tmp/tq_conn_keyword_alerts/
    ```

    b. Archive the folder with the .whl files:

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

    c. Transfer all the whl files, the connector and all the dependencies, to the ThreatQ instance.
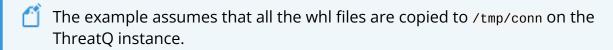
d. Open the archive on ThreatQ:
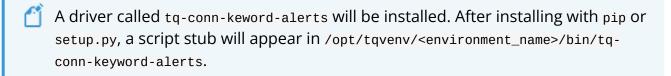
```
<> tar -xvf tq_conn_keyword_alerts.tgz
```

e. Activate the virtual environment:

```
<> source /opt/tqvenv/<environment_name>/bin/activate
```

f. 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_keyword_alerts-<version>-
   py3-none-any.whl --no-index --find-links /tmp/conn/
```

> 📝 A driver called `tq-conn-keword-alerts` will be installed. After installing with `pip` or `setup.py`, a script stub will appear in `/opt/tqvenv/<environment_name>/bin/tq-conn-keyword-alerts`.

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:

```
<> /opt/tqvenv/<environment_name>/bin/tq-conn-keyword-alerts -
   ll /var/log/tq_labs/ -c /etc/tq_labs/ -v3
```

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.
ThreatQ Client ID	This is the OAuth id that can be found at Settings Gear → User Management → API details within the user's details.

PARAMETER	DESCRIPTION
ThreatQ Username	This is the User in the ThreatQ System for integrations.
ThreatQ Password	The password for the above ThreatQ account.

Example Output

```
/opt/tqvenv/<environment_name>/bin/tq-conn-keyword-alerts-ll /var/log/tq_labs/ -c /etc/tq_labs/ -v3
ThreatQ Host: <ThreatQ Host IP or Hostname>
ThreatQ Client ID: <ClientID>
ThreatQ Username: <EMAIL ADDRESS>
ThreatQ Password: <PASSWORD>
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
Keyword List	Line break separated keywords list to generate alerts from.
Objects to Alert For	The objects to be alerted for when a keyword is detected.
Sender Email	The email addressed from which the reports will be sent.
Sender Password	The password associated with the email address.
SMTP Server	The SMTP Server associated with the email provider that will be used.
SMTP Port	The port used by the SMTP server.
Use SSL	Whether to connect to the SMTP server using SSL.
Default Recipients	Line break separated list of recipients. If none provided, the sender's email will be used.

PARAMETER	DESCRIPTION
ThreatQ Hostname/IP	ThreatQ instance's hostname or IP, in order to be link back to it.
Date Critera	Date type criteria used when searching your Threat Library (Created At; Updated At).
Email Theme	The theme used for the email's HTML (Light; Dark).

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:

```
<> /opt/tqvenv/<environment_name>/bin/tq-conn-keyword-alerts -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	Review all additional options and their descriptions.
-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.
-n, --name	Optional - Name of the connector (Option used in order to allow users to configure multiple Intelligence Mailbox connector instances on the same TQ box).
-hist, --historical {DATE}	Optional - Allows you to set the start date for the Threat Library search.

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 * * * /opt/tqvenv/<environment_name>/bin/tq-conn-
   keyword-alerts -c /etc/tq_labs/ -ll /var/log/tq_labs/ -v3
```

4. Save and exit CRON.

Known Issues / Limitations

- New or recently updated content may not be included in emails if the connector is run during the reindex process.

Change Log

- **Version 1.0.0 rev-b**
  - ○ Guide Update - Updated pathways in connector commands.
- **Version 1.0.0 rev-a**
  - ○ Guide Update - Updated virtual environment commands.
- **Version 1.0.0**
  - ○ Initial Release