ThreatQuotient

VMWare Carbon Black Cloud Enterprise EDR Connector Guide

Version 1.2.1

April 04, 2022

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

ThreatQ Supported

Support

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

Contents

Support **4**
Versioning **5**
Introduction **6**
Prerequisites **7**
Carbon Black Cloud Enterprise EDR RBAC Permissions 7
Timezones 7
PIP.conf 8
Integration Dependencies 8
Installation **9**
Creating a Python 3.6 Virtual Environment 9
Installing the Connector 10
Configuration **13**
Usage **15**
Command Line Arguments 16
CRON 17
Change Log **18**

Warning and Disclaimer

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

- Current integration version: `1.2.1`
- Supported on ThreatQ versions >= `4.34.0`
- Python: `3.6`

Introduction

The VMware Carbon Black Cloud Enterprise EDR for ThreatQ integration allows a user to export prioritized threat intelligence from ThreatQ into reports within VMware Carbon Black Cloud Enterprise EDR. VMware Carbon Black Cloud Enterprise EDR will match endpoint activity to the threat intelligence from ThreatQ and generate alerts.

VMware Carbon Black Cloud Enterprise EDR supports the following indicator types:

- IPAddress
- MD5
- SHA-256
- FQDN

The connector name and driver have changed as of version 1.2.0. In order to maintain backwards compatibility, you can use the old connector name in commands for running the connector along with the new name. All the commands used in this doc, excluding the installation steps, reference the new name `tq-conn-cb-enterprise-edr` but can be used interchangeably with the old name `tq-conn-cb-threat-hunter`.

You must use the `tq-conn-cb-threat-hunter` name when performing pip install commands.

Prerequisites

Review the following prerequisites before installing the connector.

Carbon Black Cloud Enterprise EDR RBAC Permissions

The following permissions, on the Carbon Black Cloud Enterprise EDR side, are required in order to successfully run the integration.

PERMISSION (.NOTATION NAME)	OPERATION
org.feeds	READ

Timezones

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
```

Change Time Zone Example

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

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
```

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.

The connector name and driver have changed as of version 1.2.0. In order to maintain backwards compatibility, you can use the old connector name in commands for running the connector along with the new name. All the commands used in this doc reference the new name `tq-conn-cb-enterprise-edr` but can be used interchangeably with the old name `tq-conn-cb-threat-hunter`. with the only exception being the install commands.

You must use the `tq-conn-cb-threat-hunter` name when performing a pip install/ upgrade.

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_cb_threat_hunter
   ```

   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_cb_threat_hunter

   pip download tq_conn_cb_threat_hunter -d

   /tmp/tq_conn_cb_threat_hunter/
```

b. Archive the folder with the .whl files:

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

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

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

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-cb-threat-hunter -
   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.
ThreatQ Username	This is the email address of 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-cb-threat-hunter -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
Enterprise EDR API FQDN	The FQDN to access Enterprise EDR's API. The default is `defense.conferdeploy.net`
Enterprise EDR API Secret Key	Your Enterprise EDR API Secret Key for authentication.
Enterprise EDR API ID	Your Enterprise EDR API ID for authentication.
Enterprise EDR Organization Key	Your Enterprise EDR Organization Key for authentication.
Data Collection Names (Threat Library)	A comma-separated list of Threat Library collection names you want to export.
Report Tags	A comma-separated list of tags to add to the reports. The tags will be added to each report.

PARAMETER	DESCRIPTION
ThreatQ Hostname or IP Address	This is the hostname or IP address of your ThreatQ instance in order to link back to it. This is typically the domain/IP that can be viewed in your browser's URL bar.

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

The connector name and driver have changed as of version 1.2.0. In order to maintain backwards compatibility, you can use the old connector name in commands for installing/running the connector along with the new name. All the commands used in this doc reference the old name `tq-conn-cb-threat-hunter` but can be used interchangeably with the new name `tq-conn-cb-enterprise-edr`

Use the following command to execute the driver:

```
/opt/tqvenv/<environment_name>/bin/tq-conn-cb-threat-hunter -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.
`-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).
`-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.
`-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

ARGUMENT	DESCRIPTION
	exports after the test. THIS SHOULD NEVER BE USED IN PRODUCTION.

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-cb-
   threat-hunter -c /etc/tq_labs/ -ll /var/log/tq_labs/ -v3
   ```

4. Save and exit CRON.

Change Log

- **Version 1.2.1**
  - Updated the user field validation.
- **Version 1.2.0 rev-b**
  - Guide Update - Version 1.2.0 provides several updates to the configuration parameters. If you are upgrading from a previous version, you must first delete the previous version's configuration file before proceeding with the install steps. Failure to delete the previous configuration file will result in the connector failing.
- **Version 1.2.0 rev-a**
  - Revised the installation section of the user guide. The installation section previously stated that you can use either connector name to install the integration. You must use the `tq-conn-cb-threat-hunter` name to install or upgrade the connector. All other commands can use either of the new commands: `tq-conn-cb-threat-hunter` or `tq-conn-cb-enterprise-edr`.
- **Version 1.2.0**
  - Rebranded connector name from Carbon Black Threat Hunter to VMware Carbon Black Cloud Enterprise EDR. The connector name and driver have changed as of version 1.2.0. In order to maintain backwards compatibility, you can use the old connector name in commands for installing/running it along with the new one. All the commands used in this doc reference the new name `tq-conn-cb-enterprise-edr` but can be used interchangeably with the old name `tq-conn-cb-threat-hunter`.
  - Added submission support for SHA-256 indicator types.
  - Added a new Prerequisites chapter for permissions and timezone requirements.
- **Version 1.1.0**
  - Added Python 3 support.
- **Version 1.0.0**
  - Initial Release