

ThreatQuotient



Carbon Black Protection Connector Guide

Version 1.0.0

Monday, April 6, 2020

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.

Copyright © 2020 ThreatQuotient, Inc.

All rights reserved. This document and the software product it describes are licensed for use under a software license agreement. Reproduction or printing of this document is permitted in accordance with the license agreement.

Last Updated: Monday, April 6, 2020

Contents

Warning and Disclaimer	2
Contents	3
Introduction	4
Support Matrix	4
Installation	5
With Internet access	5
Offline installation without access to the Internet	5
Executing the Driver	6
Configuration	7
CRON	9
Driver Command Line Options	10

Introduction

The Carbon Black Protection connector is used to apply policy rules to MD5, SHA-1 and SHA-256 hashes in CB Protection. The rules it applies are ban, approve or unapprove. When executed the connector will collect all hashes from saved searches configured in the ThreatQ UI, parse the IOC values and send them to CB Protection. Each saved search corresponds to the policy rules available in Carbon Black - unapprove, approve and ban.

Support Matrix

- Current integration version 1.0.0
- Supported on ThreatQ versions >= 4.33.0

Operating System	OS Version	Python Version	Notes
RedHat/CentOs	7	2.7.12	
Ubuntu	16.04	2.7.12	This has not been tested
Windows	2012R2/10	2.7.12	

Installation

There are two methods to installing the VMware Carbon Black Protection connector:

- [With Internet access](#)
- [Offline installation without access to the Internet](#)

With Internet access

This package is available in `.tar.gz` and `.whl` formats, and can be installed from the ThreatQ integrations repository. To install the `.tar.gz` or `.whl` formats:

```
$ pip install tq_conn_cb_protection
```

Offline installation without access to the Internet

When ThreatQ is deployed in an air-gapped environment without access to the internet, the whl file and its dependencies will need to be downloaded first on another machine that has internet access, transferred to the ThreatQ instance, and installed in an offline mode. This example shows the steps to do it:

```
# Download the connector whl file with its dependencies
mkdir /tmp/cb-protection
pip download tq_conn_cb_protection -d /tmp/cb-protection/

# Archive the folder with the whl files
tar -czvf cb_protection.tgz /tmp/cb-protection/
```

```
# Transfer all the whl files, the connector and all
the dependencies, to the ThreatQ instance

# Open the archive on ThreatQ
tar -xvf cb_protection.tgz

# 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_protection-1.0.0-
py2-none-any.whl --no-index --find-links /tmp/conn/
```

Executing the Driver

This package comes with a driver called `tq-conn-cb-protection`. After installing with `pip` or `setup.py` a script stub will appear in `/usr/bin/tq-conn-cb-protection`. To execute the feed just use:

```
$ tq-conn-cb-protection -c
/path/to/config/directory/ -ll
/path/to/log/directory/ -v VERBOSITY_LEVEL
```

The driver will run once, where it will connect to the TQ instance and will install the UI component of the connector. The connector must be [configured and enabled](#) in the ThreatQ platform UI.

Configuration


Navigate to the custom connectors configuration page and enter the following values for the Carbon Black Protection integration. Once completed, save the configuration by clicking on the **Save Changes** button:






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

To configure the connector:

1. Click on the **Settings** icon and select **Incoming Feeds**.
2. Locate the connector under the **Lab** tab.
3. Click on the **Feed Settings** link for the connector.
4. Under the **Connection** tab, enter the following configuration parameters:

Parameter	Description
IP/Hostname	Hostname or IP address of Carbon Black Protection.
Port	Port for communicating with the Carbon Black Protection instance. <div> The default is 443.</div>
API Key	API Key for the Carbon Black Protection instance.
Saved search with indicators to ban in Carbon Black Protection	The name of the saved search in the ThreatQ instance with indicators to be banned in Carbon Black Protection.

Parameter	Description
Saved search with indicators to approve in Carbon Black Protection	<p>The name of the saved search in the ThreatQ instance with indicators to be approved in Carbon Black Protection.</p> <div> This parameter is optional.</div>
Saved search with indicators to unapprove in Carbon Black Protection	<p>The name of the saved search in the ThreatQ instance with indicators to be unapproved in Carbon Black Protection.</p> <div> This parameter is optional.</div>
Comma-separated list of policy IDs apply to apply to this rule	<p>List of IDs of policies where this rule applies. Default is 0 which is a global rule.</p> <div> The default is 0 which is a global rule.</div>

5. Click on **Save Changes**.
6. Click on the toggle switch to the left of the connector name to enable the connector.

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 ran 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 at the top of every hour.

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

```
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:

Hourly Example

```
0 * * * * /usr/bin/tq-conn-cb-protection -c  
/path/to/config/directory/ -ll  
/path/to/log/directory/ -v VERBOSITY_LEVEL
```

4. Save and exit cron.

Driver Command Line Options

The connector's driver has several command line arguments that will help you and your customers execute this. They are listed below.

You can see these by executing:

```
/usr/bin/tq-conn-cb-protection --help
```

Output

```
usage: tq-conn-cb-protection -ll [LOGLOCATION] -c  
[CONFIG] -v [VERBOSITY_LEVEL]
```

Optional Arguments:

Argument	Description
<code>-h, --help</code>	Shows the help message and exit
<code>-ll</code> <code>LOGLOCATION,</code> <code>--log-</code> <code>location</code> <code>LOGLOCATION</code>	This sets the logging location for this connector. The location should exist and be writable by the current user. A special value of 'stdout' means to log to the console (this happens by default)
<code>-c CONFIG, -</code> <code>-config</code> <code>CONFIG</code>	This is the location of the configuration file for the connector. This location must have read and write permissions for the current user. If no config file is given, the current directory will be used. This file is also where some information from each run of the connector may be put (e.g. last run time, private OAuth, etc)

Argument	Description
<code>-cache, --cache</code>	(required) The path to the directory where you want to store your cache
<code>-v {1,2,3}, --verbosity {1,2,3}</code>	This is the logging verbosity level. The Default is 1 (Warning).