

ThreatQuotient



VMware Carbon Black EDR Connector Guide

Version 2.0.1

September 07, 2021

ThreatQuotient

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

Support

Email: support@threatq.com

Web: support.threatq.com

Phone: 703.574.9893

Contents

Versioning.....	4
Introduction.....	5
Installation	6
Configuration.....	7
ThreatQ Authentication	10
Generating OAuth Credentials.....	10
Options.....	11
Starting the Connector.....	12
Upgrading the Connector	13
Debugging the Connector.....	14
Change Log.....	15

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 © 2021 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.

Versioning

- Current Integration version: 2.0.0
- Supported on ThreatQ Versions: \geq 4.25.0

OPERATING SYSTEM	OS VERSION	PYTHON VERSION
RedHat/CentOS	6/7	2.7

Introduction



The **VMware Carbon Black EDR version 2.0.0 connector** replaced the **Carbon Black Response version 1.0.0 connector**.

The VMware Carbon Black EDR connector integration pulls Active indicators from a ThreatQ saved search and imports them into Carbon Black EDR as Threat Reports. This integration will be installed and run on the Carbon Black EDR server. The integration configuration allows you to customize what information is exported from ThreatQ into Carbon Black EDR by enabling you to specify multiple saved search names.

This Connector will receive data only from saved searches that contain Indicators of type:

IPv4, IPv6, MD5, FQDN

Installation

The VMware Carbon Black EDR Response Connector is packaged into an RPM file which will be installed on your Carbon Black EDR Response server. You can download the RPM file from either:

- ThreatQ Marketplace: <https://marketplace.threatq.com/>



Users are required to log in with their support portal credentials in order to download the integration.

- ThreatQ Download Repository: <https://download.threatq.com/integrations/>



Users are required to log in with their YUM credentials in order to download the integration from the ThreatQ Download Repository.

1. Transfer the RPM file to your Carbon Black EDR Response instance:

```
<> scp python-cb-threatq-connector-2.0.0-10.x86_64.rpm root@<cb-  
ip-address>:/tmp/
```

2. SSH into your Carbon Black EDR Response instance.

3. Install the package using the RPM:

```
<> cd /tmp  
  
rpm -ivh python-cb-threatq-connector-2.0.0-10.x86_64.rpm --  
ignoreos --nofiledigest
```



The OS is ignored in the command above. If the OS is not ignored, the installation will fail.

Configuration

The connector now needs to be configured on the Carbon Black EDR Response instance side.

1. Create a default "credentials.response" file for the connector to use.

```
<> [default]
url=https://localhost
token=xxxxxxxx
ssl_verify=False
```



Your token can be found in your user profile in the Carbon Black EDR Web UI.

2. Locate the sample of the configuration file at:

```
<> /etc/cb/integrations/threatq/connector.conf.example
```

3. Copy the sample file to the following pathway:

```
<> /etc/cb/integrations/threatq/connector.conf
```

4. Configure the ThreatQ connector by entering the following parameters:

```
<> [auth]
#-----
# ThreatQ API configuration
#-----
# This section allows global configuration options to be
# passed to the ThreatQ feed.

threatq_host=https://<your threatq host>

# Username and password and/or OAuth authentication methods
# can be configured.
# Username and password having priority if both specified.

# 1
threatq_client_id=
threatq_user=
threatq_pass=

# 2
threatq_oauth_client_id=
threatq_oauth_client_secret=
```

```
threatq_verify_ssl=false
threatq_http_proxy=
threatq_https_proxy=


[bridge]
#-----
# Core Configuration
#-----
listener_port=6300
listener_address=127.0.0.1
feed_retrieval_minutes=60

# debug=1

# API key for an administrative user of the Carbon Black EDR
server carbonblack_server_token=
carbonblack_server_sslverify=false

# Only uncomment out the carbonblack_server_url if you are
running the connector on a machine
# *other* than the Cb server itself.
# carbonblack_server_url=

# If you need to use an HTTPS proxy to access the ThreatQ API
server, uncomment and configure the https_proxy # variable
below.
# https_proxy=http://
proxyuser:proxypass@proxyhostname:proxyport
# http_proxy=http://
proxyuser:proxypass@proxyhostname:proxyport
```

PARAMETER	DETAILS
threatq_host	Example: https://<>
threatq_client_id	Client ID of ThreatQ instance specified above (on threatq_host) <div> The Client ID can be found on the UI of ThreatQ.</div>
threatq_user	Username used to connect to ThreatQ instance.
threatq_password	Password used to connect to ThreatQ instance.

**threatq_
oauth_client_id**

OAuth client id used to connect to ThreatQ instance.

**threatq_
oauth_client_secret**

OAuth client secret used to connect to ThreatQ instance.

**threatq_saved_
search_names**

This names can be found on the UI of ThreatQ instance, on the saved search section.



This is a comma-delimited list of saved searches.

threatq_verify_ssl

This is whether you want to verify the SSL connection between CB Response and ThreatQ. Setting this to true may cause connection issues.



This option is set to false by default

threatq_http_proxy

This is optional. If you want to communicate with ThreatQ via an HTTP proxy, set it here.



This must include the username, password, host/ ip, and port.

threatq_https_proxy

This is optional. If you want to communicate with ThreatQ via an HTTPS proxy, set it here.



This must include the username, password, host/ ip, and port.

ThreatQ Authentication

There are two types of authentication that can be used for connecting to the ThreatQ instance.

First one uses the `threatq_client_id`, `threatq_username`, `threatq_password` for the specified ThreatQ instance.

The second one uses OAuth (`threatq_oauth_client_id`, `threatq_oauth_client_secret`).



OAuth is supported on ThreatQ instance starting with version v4.30.

Notes:

- If fields for both authentication types are completed in the configuration file (section # 1 and # 2 located in the configuration file listed in step 4 of the [Configuration](#) chapter) then priority will have the client_id/user/password authentication, so this one will be used for connecting to the instance
- If for both section from authentication (section # 1 and # 2 located in the configuration file listed in step 4 of the [Configuration](#) chapter) fields are missing or are empty then an error will be raised when the connector service will start
- If just one of the sections from authentication (section # 1 and # 2 located in the configuration file listed in step 4 of the [Configuration](#) chapter) has all fields completed and non empty that this one will be used.

Generating OAuth Credentials



The OAuth authentication method requires ThreatQ version 4.30.

1. Run the following commands to generate the tokens:

```
<> cd /var/www/api

sudo php artisan threatq:oauth2-client --name "Carbon Black
EDR Response Connector" --user_group "analyst"
```



Save these two tokens in a safe and secure location.

Options

The server can be configured to retrieve data from a ThreatQ saved search at specific intervals.

The [bridge] section of the configuration file the parameter `feed_retrieval_minutes` must be completed in order to use this option. See [step 4 in the Configuration chapter](#) for more details.

Starting the Connector

Prerequisites

- Connector has been installed via RPM.
- Connector has been configured on the ThreatQ side (created the saved searches).
- Connector has been configured on the Carbon Black EDR Response side (connector.conf).

Run one of the following commands, depending on the CentOS version:

Centos6

```
<> service cb-threatq-connector start
```

Centos7

```
<> systemctl start cb-threatq-connector
```

Upgrading the Connector

If the ThreatQ connector has already been installed once and needs to be updated, follow these steps:

1. Transfer the new RPM file to your Carbon Black EDR Response instance:

```
<> scp python-cb-threatq-connector-2.0.0-10.x86_64.rpm root@:/tmp/
```

2. SSH into your Carbon Black EDR Response instance.
3. Stop the old connector:

```
<> service cb-threatq-connector stop
```

4. Use the RPM to upgrade your connector:

```
<> cd /tmp rpm -Uvh --force python-cb-threatq-connector2.0.0-10.x86_64.rpm --ignoreos --nofiledigest
```

5. Start the connector up again:

```
<> service cb-threatq-connector start
```

Debugging the Connector

If the connector is up and running, but you aren't seeing any indicators within Carbon Black EDR Response, you can debug it by looking at the connector log file:

Log Location: `/var/log/cb/integrations/cb-threatq-connector/cbthreatq-connector.log`

If you want to see the data that is being downloaded by Carbon Black EDR Response, you can directly curl the connector from your Carbon Black EDR Response instance.

1. SSH into your Carbon Black EDR Response instance.
2. Use curl to send a request to the connector

```
<> curl -k http://127.0.0.1:6300/threatq/json?server_  
token=<enter your server token here>
```



You can get the server token from the log file specified above. Otherwise, you might just be able to leave it out.

Change Log

- **Version 2.0.1**
 - Rebuilt RPM using ThreatQ SDK 1.8.1.
- **Version 2.0.0**
 - Initial Release - Replaced Carbon Black Response Connector