

ThreatQuotient



TheHive Connector User Guide

Version 1.3.0 rev-a

July 19, 2023

ThreatQuotient

20130 Lakeview Center Plaza Suite 400
Ashburn, VA 20147

 **ThreatQ Supported**

Support

Email: support@threatq.com

Web: support.threatq.com

Phone: 703.574.9893

Contents

Warning and Disclaimer	3
Support	4
Integration Details.....	5
Introduction	6
Prerequisites	7
Hive Requirements	7
Time Zone	7
Integration Dependencies	8
Installation.....	9
Configuration	12
Enabling Webhooks on The Hive	13
Usage.....	15
Command Line Arguments.....	16
Setting up the Integration as a Service.	17
Running with Gunicorn	18
Gunicorn Command Line Arguments.....	18
Installing The Hive Training VM	20
Change Log	21

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 © 2023 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.

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.

Integration Details

ThreatQuotient provides the following details for this integration:

Current Integration Version 1.3.0

**Compatible with ThreatQ
Versions** $\geq 4.40.0$

The Hive Version $\geq 4.0.0$

Python Version 3.6

Support Tier ThreatQ Supported

Introduction

The Hive custom connector for ThreatQ is a server that responds to new cases created, and edits to existing cases in real time. The connector then pushes that data to ThreatQ for analysis.



The integration is required to be installed on The Hive VM.

Prerequisites

Review the following requirements before attempting to install the connector.

Hive Requirements

Confirm that you have the following Hive information available before starting the installation process:

- Ip Address/Hostname for Hive
- The port TheHive is running on, if not port 80/443
- API Key

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
```

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	>= 1.8.0	N/A
threatqcc	>= 1.4.0	N/A
wheel	N/A	N/A
flask	0.12.5	Pinned Version

Installation

The following provides you with steps on installing the connector in a python 3 environment on a Hive VM.



The integration is required to be installed on The Hive VM.



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. The connector can be installed from the ThreatQuotient repository with YUM credentials or offline via a .whl file.
2. Install the connector using one of the following methods:

ThreatQ Repository

- a. Run the following command:

```
python3 -m pip install -i https://  
<USERNAME>:<PASSWORD>@extensions.threatq.com/threatq/integrations  
tq_mw_thehive
```

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 Hive instance.

- a. Download the connector whl file with its dependencies:

```
mkdir /tmp/tq_mw_thehive  
  
pip download -i https://  
<USERNAME>:<PASSWORD>@extensions.threatq.com/threatq/  
integrations -d /tmp/tq_mw_thehive/
```

- b. Archive the folder with the .whl files:

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

- c. Transfer all the whl files, the connector and all the dependencies, to the Hive instance.
- d. Open the archive:

```
tar -xvf tq_mw_thehive.tgz
```

- e. Install the connector:



The example assumes that all the whl files are copied to /tmp/conn/ on the Hive instance.

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

A driver called `tq-mw-thehive` will be installed.

3.



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/
```

4. Perform an initial run using the following command:

```
tq-mw-thehive -ll /var/log/tq_labs/ -c /etc/tq_labs/ -v3
```

5. 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 CID (Client ID)	This is the OAuth ID that can be found under the ThreatQ user profile by navigating to the Systems gear icon > User Management and clicking the user.
Email	The username that you use to log into ThreatQ. This should be a Maintenance or Administrative account.
Password	The password associated with the username above.
Status	The default status for IoCs that are created by this integration. It is common to set this to Active but organization SOPs should be respected when setting this field.

PARAMETER	DESCRIPTION
TheHive Hostname/IP	This is the host of TheHive instance. either the IP Address or hostname as resolvable by the system this integration is installed on, as well as the port if necessary.
TheHive API Key	The API Key used to authenticate with The Hive.
Sync indicators only if is IOC field is selected?	Whether or not the user only wants to sync indicators if the is_ioc field is selected in The Hive.
Sync cases as Incidents instead of Events?	Whether or not the user wants The Hive cases to sync as Incidents or Events.

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 entry to open its details page.
4. Enter the following parameters under the **Configuration** tab:

PARAMETER	DESCRIPTION
TheHive Hostname/ IP	This is the host of TheHive instance. either the IP Address or hostname as resolvable by the system this integration is installed on, as well as the port if necessary.
The Hive API Key	The API Key used to authenticate with The Hive.
Sync indicators if is IOC field is selected?	Whether or not the user only wants to sync indicators if the <code>is_ioc</code> field is selected in The Hive.
Sync cases as Incidents instead of Events?	Whether or not the user wants The Hive cases to sync as Incidents or Events.
Sync cases with no observables?	Select whether to sync cases that have no observables attached.

5. Review any additional settings, make any changes if needed, and click on **Save**.
6. Click on the toggle switch, located above the *Additional Information* section, to enable it.

Enabling Webhooks on The Hive

In order to forward events from The Hive to the integration, you first have to enable webhooks on the Hive server.

1. SSH to The Hive console.
2. Open the configuration file for The Hive in the editor of your choice. The file is typically located at `/etc/thehive/application.conf`.
3. Add the webhook config at the bottom of the configuration file. An example of this is below.

```
notification.webhook.endpoints = [  
  {  
    name: ThreatQ  
    url: "http://127.0.0.1:5000/cases"  
    version: 0  
    wsConfig: {}  
    includedTheHiveOrganisations: []  
    excludedTheHiveOrganisations: []  
  }  
]
```

4. Save the file.
5. Restart The Hive service:

```
systemctl restart thehive
```

6. Use the following commands to save the username/password needed to enable the webhook config. This will hide the credentials so that they won't be logged.

```
read -p 'Enter the URL of TheHive: ' thehive_url  
read -p 'Enter your login: ' thehive_user  
read -s -p 'Enter your password: ' thehive_password
```

7. Then make the request to the Hive server.

```
curl -XPUT -u$thehive_user:$thehive_password -H 'Content-type: application/  
json' $thehive_url/api/config/organisation/notification -d '  
{  
  "value": [  
    {  
      "delegate": false,  
      "trigger": { "name": "AnyEvent"},  
      "notifier": { "name": "webhook", "endpoint":  
"ThreatQ" }  
    }  
  ]  
}'
```

You can use the following request to verify that the configuration has been saved on The Hive:

```
curl -XGET -u$thehive_user:$thehive_password -H 'Content-type: application/
json' $thehive_url/api/config/organisation/notification
```

Usage

Once the connector is installed in the ThreatQ UI and enabled, you will re-run the Initial Configuration command in order to kick off the integration.

```
tq-mw-thehive -ll /var/log/tq_labs -c /etc/tq_labs -v3
```

Once the integration successfully completes, it is recommended that you set up the integration to [run as a service](#) on the Hive.

Command Line Arguments

This connector supports the following custom command line arguments:

ARGUMENT	DESCRIPTION
<code>-h, --help</code>	Shows the help message and exits.
<code>-v, --verbosity {1,2,3}</code>	Sets the log verbosity level (3 means everything). A special value of 'stdout' means to log to the console (this happens by default)
<code>-c, --config</code>	The path to the directory where you want to store your config file. 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.)
<code>-ll, --loglocation</code>	The path to the directory where you want to store your logs
<code>-ep, --external-proxy</code>	This allows you to use the proxy that is specified in the ThreatQ UI
<code>-p, --port</code>	The port for TheHive WebHooks server to run on. Default port number is 5000

Setting up the Integration as a Service.

Setting up the integration as a service ensures that the app is always running and can send tickets in real time to the ThreatQ appliance. This also ensures that the application gets started back up in the event of a server reboot.

On the The Hive instance:

1. Create a new file for the service called: `/etc/systemd/system/tq_thehive.service`.
2. Copy and paste the following into the file:

```
[Unit]
Description=TheHive websockets server for ThreatQ
[Service]
Type=simple
User=root
WorkingDirectory=/root
ExecStart=/usr/bin/tq-mw-thehive -c /etc/tq_labs/ -ll /var/log/tq_labs/
-v3
Restart=always
TimeoutSec=10
[Install]
WantedBy=multi-user.target
```

3. Save the file.
4. Reload the services:

```
systemctl daemon-reload
```

5. Enable the service:

```
systemctl enable tq_thehive.service
```

6. Start the service:

```
systemctl start tq_thehive.service
```



Optionally, you can add the `-p {{port}}` or `--port {{port}}` flag to the `ExecStart` line of the service file, to keep the integration running on your desired port.

Running with Gunicorn

You can run the application with Gunicorn using a wsgi instead of running it in a service. Run the following command:

```
gunicorn -b 0.0.0.0:5000 'tq_mw_thehive.tq_driver:run()'
```

Gunicorn Command Line Arguments

FLAG	DESCRIPTION	EXAMPLE
<code>-b</code>	The host and port that you want to bind Gunicorn to listen on.	<code>-b 0.0.0.0:5000</code>
<code>-w</code>	This determines how many worker threads Gunicorn will use for the application.	<code>-w 4</code>
<code>--log-level</code>	This is the amount of information you will see from logs. Set this argument to debug to see all information.	<code>--log-level=debug</code>
<code>--log-file</code>	This is the location of the Gunicorn log file.	<code>--log-file=/var/log/tq_labs/thehive.log</code>
<code>-t</code>	This is the timeout for how long it takes for the application to respond in Gunicorn. Setting this to 0 gives an unlimited time out.	<code>-t 0</code>

Some arguments can be used by the application. For instance the default configuration location is set to `/etc/tq_labs`. This will read the configuration out of the `/etc/tq_labs/` directory by default.

Logs, by default, will be added to the Gunicorn logs and will not be recorded in the normal log location. Arguments can be passed to the integration itself using the examples listed below.

FLAG	DESCRIPTION	EXAMPLE
<code>-c</code>	This is the location of the configuration file for the connector.	<code>'tq_mw_thehive.tq_driver:run(c="/etc/tq_labs/")'</code>

FLAG	DESCRIPTION	EXAMPLE
-n	This is the name of the connector.	'tq_mw_thehive.tq_driver:run(n="TheHive")'

Installing The Hive Training VM

To install the integration on The Hive's training vm, there is a process that needs to take place before you can install and run the connector.

1. Elevate to the root user:

```
sudo su
```

2. Change to the root user directory:

```
cd ~
```

3. Change the umask:

```
umask 022
```

4. Return to the logged in user:

```
exit
```

5. Install the integration:

```
python3 -m pip install -i https://  
<USERNAME>:<PASSWORD>@extensions.threatq.com/threatq/integrations  
tq_mw_thehive
```

Change Log

- **Version 1.3.0 rev-a**
 - Corrected a command in the Setting up the Integration as a Service chapter.
- **Version 1.3.0**
 - Added capability to choose to sync cases that have no observables attached to them.
 - Added capability to only sync cases that have observables which are IOCs.
- **Version 1.2.0**
 - Added the ability to run the integration with Gunicorn.
- **Version 1.1.0**
 - Added the ability to run the integration on a different port other than the default port of 5000.
 - Added the ability to only sync indicators if the 'is ioc' field in the hive is selected.
 - Added the ability to sync cases as incidents instead of events.
- **Version 1.0.0**
 - Initial release.