ThreatQuotient

The Hive Connector Guide

Version 1.2.0 rev-a

March 05, 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

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.

Versioning

- Current integration version: `1.2.0`
- Supported on ThreatQ versions >= `4.40`
- Category: `Labs`
- The Hive version: `>=4.0`
- Python version: `>=3.6`

Introduction

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

Prerequisites

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

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

Installation

The integration should be installed on The Hive VM.

Upgrading - If you are upgrading from a previous version, review the Change Log to determine if there are any changes to configuration file via new or removed fields. If there are changes, you must first delete your existing configuration file before proceeding with the steps below to install the new version.

If you need assistance, please open a support ticket with ThreatQ Support.

1. Install the connector using one of the following methods:

   ThreatQ Repository

   This package is available in a `.whl` format, and can be installed from the ThreatQ integrations repository.

   a. To install it run the following command:

   ```
   python3 -m pip install 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 ThreatQ instance.

   a. Download the connector whl file with its dependencies:

   ```
   /tmp/tq_mw_thehive

   pip download tq_mw_thehive -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 ThreatQ instance.

   d. Open the archive on ThreatQ:

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

e. Install the connector on the ThreatQ instance.

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

```
python3 -m pip install /tmp/conn/tq-mw-thehive-<version>-
py3-none-any.whl --no-index --find-links /tmp/conn/
```

A driver called `tq-mw-thehive` is now installed.

2. Create the configuration and log directories using the following commands:

```
mkdir -p /etc/tq_labs
mkdir -p /var/log/tq_labs
```

3. Perform the initial run using the following command:

```
tq-mw-thehive -c /var/log/tq_labs -ll /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 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 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 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.

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.

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 wont 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.

> Once the integration successfully completes, you will need to set up a CRON-job for it so it can run on a schedule. You can optionally set up the integration to run as a service.

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

Command Line Arguments

ARGUMENT	DESCRIPTION
`-h`, `--help`	Shows the help message and exits.
`-v`, `--verbosity {1,2,3}`	Sets the log verbosity level (3 means everything). A special value of 'stdout' means to log to the console (this happens by default)
`-c`, `--config`	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.)
`-ll`, `--loglocation`	The path to the directory where you want to store your logs
`-ep`, `--external-proxy`	This allows you to use the proxy that is specified in the ThreatQ UI

ARGUMENT	DESCRIPTION
`-p`, `--port`	The port for TheHive WebHooks server to run on. Default port number is `5000`

> All location-based options default to the current working directory if they are not provided. To find additional options and option descriptions, invoke the program with '-h'.

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 hour.

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 Hour Example

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

4. Save and exit CRON.

Setting up the Integration as a Service.

This step is optional, but it ensures that the app is always running and can send tickets in real time to the ThreatQ appliance. This will also ensure that the application gets started back up in the event of a server reboot

1. Create a new file for the service:

   ```
   touch /etc/system/system/tq_thehive.service
   ```

2. Open the file with your editor of choice.
3. Add the following to the new 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
   ```

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

   ```
   systemctl daemon-reload
   ```

6. Enable the service:

   ```
   systemctl enable tq_thehive.service
   ```

7. 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
`-b`	The host and port that you want to bind Gunicorn to listen on.	-b 0.0.0.0:5000
`-w`	This determines how many worker threads Gunicorn will use for the application.	-w 4
`--log-level`	This is the amount of information you will see from logs. Set this arugment to **debug** to see all information.	--log-level=debug
`--log-file`	This is the location of the Gunicorn log file.	--log-file=/var/log/tq_labs/thehive.log
-t	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.	-t 0

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 to the integration itself using the examples listed below.

FLAG	DESCRIPTION	EXAMPLE
`-c`	This is the location of the configuration file for the connector.	'tq_mw_thehive.tq_driver:run(c="/etc/tq_labs/")'
`-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 1.2.0 rev-a**
  - Guide Update - Corrected a typo in the initial run command in the Installation chapter.
- **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.