

# ThreatQuotient

A Securonix Company



## The Hive App

Version 1.5.0

July 15, 2025

## ThreatQuotient

20130 Lakeview Center Plaza Suite 400  
Ashburn, VA 20147

 ThreatQ Supported

## Support

Email: [support@threatq.com](mailto:support@threatq.com)

Web: [support.threatq.com](https://support.threatq.com)

Phone: 703.574.9893

# Contents

<b>Warning and Disclaimer .....</b>	<b>3</b>
<b>Support .....</b>	<b>4</b>
<b>Integration Details.....</b>	<b>5</b>
<b>Introduction .....</b>	<b>6</b>
<b>Prerequisites .....</b>	<b>7</b>
Hive Requirements .....	7
Integration Dependencies .....	8
<b>Installation.....</b>	<b>9</b>
<b>Configuration .....</b>	<b>11</b>
Enabling Webhooks on The Hive .....	13
<b>ThreatQ Mapping.....</b>	<b>15</b>
Webhook Content for Case Updates .....	15
Webhook Content for Observable Updates.....	17
Observables Type Mapping.....	18
<b>Usage.....</b>	<b>20</b>
Command Line Arguments.....	21
Running with Gunicorn .....	22
Gunicorn Command Line Arguments.....	22
Setting up the Integration as a Service with Gunicorn.....	24
<b>Installing The Hive Training VM.....</b>	<b>26</b>
<b>Change Log .....</b>	<b>27</b>

# 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 © 2025 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](mailto: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** 1.5.0

**Compatible with ThreatQ  
Versions**  $\geq 5.29.0$

**The Hive Version**  $\geq 5.0.0$

**Python Version** 3.6

**Support Tier** ThreatQ Supported

# Introduction

The Hive app for ThreatQ is a server that responds to new cases created and edits to existing cases in real time. The app 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 app.

## Hive Requirements


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

- IP Address/Hostname for Hive.
- A TheHive account with `org-admin` or `analyst` role.
- TheHive API Key associated with the account above.

## 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	>= 1.8.1	N/A
threatqcc	>= 1.4.4	N/A
<b>flask</b>	<b>== 2.0.3</b>	<b>Pinned</b>
<b>setuptools</b>	<b>== 59.6.0</b>	<b>Pinned</b>
<b>jinja2</b>	<b>== 3.0.3</b>	<b>Pinned</b>
<b>Werkzeug</b>	<b>== 2.0.3</b>	<b>Pinned</b>
wheel	N/A	<b>Pinned</b>
gunicorn	N/A	Only required if the integration is <a href="#">run as a service with gunicorn</a> .



# Installation

The following provides you with steps on installing the app 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 app failing.

1. The app can be installed from the ThreatQuotient repository with YUM credentials or offline via a .whl file.
2. Install the app 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 app from a wheel file, the wheel file (.whl) will need to be copied via SCP into your Hive instance.

- a. Download the app 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 app and all the dependencies, to the Hive instance.
- d. Open the archive:

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

- e. Install the app:



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.
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.

You will still need to [configure and then enable the app](#).



# 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 URL	<p>Enter the URL of your TheHive instance.</p> <div>  <p>The format should be: <code>https://&lt;hostname&gt;:&lt;port&gt;</code>. Only include the port if applicable.</p> </div>
The Hive API Key	Enter your API Key used to authenticate with The Hive.
Only sync indicators if the 'Is IOC' field is selected	<p>Enable this parameter to sync only observables that are marked as an IOC in TheHive. This parameter is disabled by default.</p> <div>  <p>All observables are synced by default.</p> </div>
Sync cases as Incidents instead of Events?	Enable this parameter to sync cases as ThreatQ incidents. Otherwise, cases will sync as ThreatQ events.
Include cases with no observables?	Enable this parameter to only sync cases containing at least one observable.
Use Custom IoC Mapping	Enable this parameter to customize how observables are ingested into ThreatQ. By default IoCs are ingested according to the documentation. ( <i>default: False</i> )

PARAMETER	DESCRIPTION
The Hive IoC Type Mapping	<p>Specify what observables should be ingested into ThreatQ. Enter a line for each observable type having the format:&lt;Observable Type&gt;: &lt;ThreatQ Indicator Type&gt;. Use hash to determine ThreatQ Indicator Type by length.</p> <div>  This parameter is only accessible when the <b>Use Custom IoC Mapping</b> parameter is enabled.         </div> <p>The default value for this field is:</p> <pre> autonomous-system: ASN domain: FQDN fqdn: FQDN ip: IP Address filename: Filename mail: Email Address mail-subject: Email Subject registry: Registry Key url: URL uri_path: URL Path user-agent: User-agent hash: hash </pre>

- Review any additional settings, make any changes if needed, and click on **Save**.
- 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. There are two ways to do this:

### Via the ThreatQ User Interface:

1. Navigate to the endpoint configuration menu under **Organization**.
2. Create a new webhook connector with these values:

```
Name: ThreatQ +URL: http://127.0.0.1:5000/cases +Version: 0
```

3. Navigate to the notifications configuration menu under **Organization**.
4. Create a new notifier with these values:

```
Trigger: AnyEvent +Enable Notification: True
```

5. Add the previously created ThreatQ webhook under the 'notifiers' section of the notifier config menu.

### Via the Command Line Interface (CLI):

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. See the following as an example:

```
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
```

# ThreatQ Mapping

## Webhook Content for Case Updates

```
{
  "_id": "~43528360",
  "_type": "Audit",
  "_createdBy": "thehive@thehive.local",
  "_createdAt": 1751886507808,
  "action": "update",
  "requestId": "be8e274255641dd4:2049058:196d7e72e35:-8000:53683",
  "rootId": "~43872472",
  "details": {
    "severity": 3
  },
  "objectId": "~43872472",
  "objectType": "Case",
  "object": {
    "_id": "~43872472",
    "_type": "Case",
    "_createdBy": "thehive@thehive.local",
    "_updatedBy": "thehive@thehive.local",
    "_createdAt": 1751457012174,
    "_updatedAt": 1751886507804,
    "number": 5984,
    "title": "Hamsters Test",
    "description": "Test integration with ThreatQ",
    "severity": 3,
    "severityLabel": "HIGH",
    "startDate": 1751456989469,
    "endDate": 1751885565331,
    "tags": [
      "utility"
    ],
    "flag": false,
    "tlp": 1,
    "tlpLabel": "GREEN",
    "pap": 1,
    "papLabel": "GREEN",
    "status": "New",
    "stage": "New",
    "summary": "dev",
    "impactStatus": "NoImpact",
    "assignee": "thehive@thehive.local",
    "customFields": [],
    "userPermissions": [],
    "extraData": {}
  }
}
```

```

    "newDate": 1751457012166,
    "inProgressDate": 1751885513068,
    "closedDate": 1751885565333,
    "timeToDetect": 22697,
    "timeToTriage": 428500902,
    "timeToAcknowledge": 428523599,
    "timeToResolve": 52263,
    "handlingDuration": 52265,
    "customFieldValues": {}
  },
  "organisation": {
    "organisationId": "~4980800",
    "organisation": "demo"
  }
}

```

ThreatQuotient provides the following default mapping from this webhook based on the `.object` content:

FEED DATA PATH	THREATQ ENTITY	THREATQ OBJECT TYPE OR ATTRIBUTE KEY	PUBLISHED DATE	EXAMPLES	NOTES
<code>._id</code> , <code>.title</code>	Event Title/ Incident Value	Event/Incident	N/A	Hive Case # ~43872472 Hamsters Test	Prepended with Hive Case #. Ingested according to used config Sync cases as Incidents instead of Events.
<code>.description</code>	Event/Incident Description	N/A	N/A	The integration with ThreatQ	N/A
<code>.tags</code>	Event/Incident Tag	N/A	N/A	utility	N/A
<code>.tlpLabel</code>	Event/Incident TLP	N/A	N/A	GREEN	N/A
<code>._id</code>	Event/Incident Attribute	TheHive Id	N/A	~43872472	N/A
<code>.status</code>	Event/Incident Attribute	Status	N/A	New	Updatable
<code>.assignee</code>	Event/Incident Attribute	Owner	N/A	thehive@thehive.local	Updatable
<code>.number</code>	Event/Incident Attribute	Case Id	N/A	5984	Updatable
<code>.severityLabel</code>	Event/Incident Attribute	Severity	N/A	GREEN	Updatable
<code>.papLabel</code>	Event/Incident Attribute	PAP	N/A	GREEN	Updatable
<code>.flag</code>	Event/Incident Attribute	Flagged	N/A	No	Updatable. Converted to Yes or No.



## Webhook Content for Observable Updates

```
{
  "_id": "~43532456",
  "_type": "Audit",
  "_createdBy": "thehive@thehive.local",
  "_createdAt": 1751887430166,
  "action": "update",
  "requestId": "be8e274255641dd4:2049058:196d7e72e35:-8000:53823",
  "rootId": "~43872472",
  "details": {
    "sighted": true
  },
  "objectId": "~85033200",
  "objectType": "Observable",
  "object": {
    "_id": "~85033200",
    "_type": "Observable",
    "_createdBy": "thehive@thehive.local",
    "_updatedBy": "thehive@thehive.local",
    "_createdAt": 1751876750526,
    "_updatedAt": 1751887430161,
    "dataType": "hash",
    "data": "ff7b2c3938306261881c42e78d0df51d9bcdd574",
    "startDate": 1751876750526,
    "tlp": 2,
    "tlpLabel": "AMBER",
    "pap": 2,
    "papLabel": "AMBER",
    "tags": [
      "ioc"
    ],
    "ioc": false,
    "sighted": true,
    "reports": {},
    "message": "unknown signature",
    "extraData": {},
    "ignoreSimilarity": false
  },
  "organisation": {
    "organisationId": "~4980800",
    "organisation": "demo"
  }
}
```

ThreatQ provides the following default mapping from this webhook based on the `.object` content:

FEED DATA PATH	THREATQ ENTITY	THREATQ OBJECT TYPE OR ATTRIBUTE KEY	PUBLISHED DATE	EXAMPLES	NOTES
<code>.data</code>	Indicator Value	<code>.dataType</code>	N/A	ff7b2c3938306261881c42e78d0df51d9bcdd574	Type determined using the table below.

FEED DATA PATH	THREATQ ENTITY	THREATQ OBJECT TYPE OR ATTRIBUTE KEY	PUBLISHED DATE	EXAMPLES	NOTES
.message	Indicator Description	N/A	N/A	unknown siganture	N/A
.tags	Indicator Tag	N/A	N/A	ioc	N/A
.tlpLabel	Indicator TLP	N/A	N/A	AMBER	N/A
.papLabel	Indicator Attribute	PAP	N/A	AMBER	Updatable
.sighted	Indicator Attribute	Sighted?	N/A	No	Updatable. Converted to Yes or No.
.ioc	Indicator Attribute	Indicator Of Compromise?	N/A	Yes	Added only if .ioc is True. Value is always Yes.

## Observables Type Mapping

The following mapping will be used if no Custom IoC Mapping is provided in the user configuration the observables from TheHive:

THE THE HIVE TYPE	THREATQ INDICATOR TYPE
autonomous-system	ASN
ip	IP Address
hash	MD5
hash	SHA-1
hash	SHA-256
hash	SHA-512
hash	SHA-384
fqdn	FQDN
url	URL

---

THE THE HIVE TYPE	THREATQ INDICATOR TYPE
-------------------	------------------------

uri_path	URL Path
----------	----------

mail	Email Address
------	---------------

mail-subject	Email Subject
--------------	---------------

filename	Filename
----------	----------

registry	Registry Key
----------	--------------

user-agent	User-agent
------------	------------



Hash type is determined using the length.

## Usage

Once the app 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
```

The above method will run the program, but for production use, it is recommended to run it with gunicorn. For more stability, create and run the integration as a service with gunicorn. Instructions to do so are below.

## Command Line Arguments

This app 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 app 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
<code>-ds, --disable-ssl</code>	Disable SSL Verification
<code>-li, --listener-ip</code>	The IP that the flask app uses to listen on. Default IP is 127.0.0.1

## 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 <b>debug</b> 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 <b>0</b> 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 app.	<code>'tq_mw_thehive.tq_driver:run(c="/etc/tq_labs/")'</code>

---

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

## Setting up the Integration as a Service with Gunicorn

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:

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

2. Open the file with your editor of choice.
3. Copy and paste the following into the file:

```
[Unit]
Description=TheHive websockets server for ThreatQ
[Service]
Type=simple
User=thehive
WorkingDirectory=/opt/thp
ExecStart=/opt/thp/.local/bin/gunicorn -b 0.0.0.0:5000
'tq_mw_thehive.tq_driver:run()' --log-file=/var/log/tq_labs/
the_hive_app.log --log-level=debug
Restart=always
TimeoutSec=10
[Install]
WantedBy=multi-user.target
```



Service User and WorkingDirectory in above example assumes that it is running as user thehive with working directory of /opt/thp. You may need to update these to fit your environment. Using which gunicorn can help to find path for ExecStart.

Optionally, you can add flags from the [Running with Gunicorn](#) section to the ExecStart line of the service file, or change the listening IP Address or Port for further customization.

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
```

# 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 app.

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 1.5.0**
  - Updated the app to utilize API version 1 endpoints.
  - Resolved an issue where a Hive instance with a large number of cases/indicators would cause the integration to fail.
  - Updated dependencies utilized by the app.
  - TLP is now set as a property of the source opposed to being ingested as an attribute.
  - All attributes can now be updated.
  - The following configuration CLI install prompts, previously set during installation, are now set in the UI configuration page:
    - **Status**
    - **TheHive Hostname/IP**
    - **TheHive API Key**
    - **Sync Indicators Only If the IOC field is selected**
    - **Sync Cases as Incidents instead of Events**
  - Added the following new configuration parameters:
    - **Use Custom IoC Mapping** - enable this parameter to customize how observables are ingested into ThreatQ.
    - **The Hive IoC Type Mapping** - specify what observables to ingest if utilizing custom IoC mapping.
  - Updated the Usage chapter and added a new section: **Setting up the Integration as a Service with Unicorn.**
  - Updated the minimum ThreatQ version to 5.29.0.
- **Version 1.4.0**
  - Added support for The Hive 5.x.
  - Renamed to The Hive App.
- **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 Unicorn.
- **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.