ThreatQuotient

Installing Custom Connectors on Another Instance

Version 1.0.1

August 09, 2024

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

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.

Installing Custom Connectors on Another Instance

Download PDF version

The following steps are for installing the custom connector on another instance other than your ThreatQ instance.

Connector Installation

1. Navigate to the ThreatQ Marketplace and download the .whl file for the integration.
2. SSH into the instance you will install the connector on.
3. Create your python 3 virtual environment:

   ```
   python3.6 -m venv /opt/tqvenv/<environment_name>
   ```

4. Active the new environment:

   ```
   source /opt/tqvenv/<environment_name>/bin/activate
   ```

5. Run the pip upgrade command:

   ```
   pip install --upgrade pip
   ```

6. Install the required dependencies:

   ```
   pip install setuptools==59.6.0 threatqsdk threatqcc
   ```

   > Some connectors may require additional dependencies. See the connector's user guide further details.

7. Activate the virtual environment if you haven't already:

   ```
   source /opt/tqvenv/<environment_name>/bin/activate
   ```

8. Transfer the whl file to the `/tmp` directory on your instance.
9. Install the connector on your instance:

   ```
   pip install /tmp/tq_conn_<wheel_name>-<version>-py3-none-any.whl
   ```

> A driver called `<connector-driver-name>` will be installed. After installing, a script stub will appear in `/opt/tqvenv/<environment_name>/bin/<connector-driver-name>`.

10. 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:

    Configuration Directory

    ```
    mkdir -p /etc/tq_labs/
    ```

    Log Directory

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

11. Perform an initial run using the following command:

    ```
    /opt/tqvenv/<environment_name>/bin/<connector-driver-name> -ll /var/
    log/tq_labs/ -c /etc/tq_labs/ -v3
    ```

12. 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 Client ID	This is the OAuth id that can be found at Settings Gear → User Management → API details within the user's details.
ThreatQ Username	This is the Email Address of the user in the ThreatQ System for integrations.
ThreatQ Password	The password for the above ThreatQ account.
Status	This is the default status for objects that are created by this Integration.

Example Output

```
/opt/tqvenv/<environment_name>/bin/tq-conn-tcl-connect -ll /var/log/
tq_labs/ -c /etc/tq_labs/ -v3
ThreatQ Host: <ThreatQ Host IP or Hostname>
ThreatQ Client ID: <ClientID>
```

```
ThreatQ Username: <EMAIL ADDRESS>
ThreatQ Password: <PASSWORD>
Status: Review
Connector configured. Set information in UI
```

Usage

Use the following command to execute the driver:

```
/opt/tqvenv/<environment_name>/bin/<connector-driver-name> -v3 -ll /var/
log/tq_labs/ -c /etc/tq_labs/
```

Command Line Arguments

This connector supports the following custom command line arguments:

ARGUMENT	DESCRIPTION
`-h`, `--help`	Review all additional options and their descriptions.
`-ll LOGLOCATION`, `--loglocation LOGLOCATION`	Sets the logging location for the connector. The location should exist and be writable by the current. A special value of 'stdout' means to log to the console (this happens by default).
`-c CONFIG`, `--config CONFIG`	This is the location of the configuration file for the connector. 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.)
`-v {1,2,3}`, `--verbosity {1,2,3}`	This is the logging verbosity level where **3** means everything.
`-n, --name`	Optional - Name of the connector (Option used in order to allow users to configure multiple connector instances on the same TQ box).

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 two hours.

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 2 Hours Example

   ```
   0 */2 * * * /opt/tqvenv/<environment_name>/bin/<connector-driver-
   name> -c /etc/tq_labs/ -ll /var/log/tq_labs/ -v3
   ```
4. Save and exit CRON.

Uninstalling the Connector

Perform the following steps to uninstall the connector and remove its CRONtab entry:

1. SSH into your instance
2. Activate the Python environment

   ```
   source /opt/tqvenv/environment_name/bin/activate
   ```
3. Uninstall the connector

   ```
   pip uninstall tq-conn-<connector-driver-name>
   ```
4. Open the Crontab

   ```
   crontab -e
   ```
5. Remove the CRON entry for the connector, save, and exit CRON. (edited)