ThreatQuotient

Darktrace Connector Guide

Version 1.0.0 rev-a

January 06, 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

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	1.0.0
Compatible with ThreatQ Versions	>= 4.55.0
Python Version	3.6
Support Tier	ThreatQ Supported
ThreatQ Marketplace	https://marketplace.threatq.com/details/darktrace-connector

Introduction

The Darktrace Connector ingests alerts from Darktrace into ThreatQ as AI Analyst and Model Breach Events with devices as related indicators and assets. It also takes a comma-separated list of Threat Library collections in as a parameter and exports the collections' FQDN and IP Address indicators into Darktrace.

Prerequisites

Review the following requirements before attempting to install the connector.

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
```

Asset Object

The integration requires the Asset object. The Asset installation files are included with the integration download on the ThreatQ Marketplace. The Asset object must be installed prior to installing the integration.

> ⚠ You do not have to install the Asset object if you are running ThreatQ version 5.10.0 or greater as the object has been seeded as a default system object.

Use the steps provided to install the Asset custom object.

> ⚠ When installing the custom objects, be aware that any in-progress feed runs will be cancelled, and the API will be in maintenance mode.

1. Download the custom object zip file from the ThreatQ Marketplace and unzip its contents.
2. SSH into your ThreatQ instance.
3. Navigate to tmp directory:

   ```
   cd /tmp/
   ```

4. Create a new directory:

   ```
   mkdir darktrace
   ```

5. Upload the **asset.json** and **install.sh** script into this new directory.
6. Create a new directory called **images** within the microsoft_365_defender_cdf directory.

   ```
   mkdir images
   ```

7. Upload the asset.svg.
8. Navigate to the **/tmp/darktrace**.

   The directory should resemble the following:

   - tmp
     - **darktrace**
       - **asset.json**
       - **install.sh**
       - **images**
         - **asset.svg**

9. Run the following command to ensure that you have the proper permissions to install the custom object:

   ```
   chmod +x install.sh
   ```

10. Run the following command:

    ```
    sudo ./install.sh
    ```

    > You must be in the directory level that houses the install.sh and json files when running this command.

The installation script will automatically put the application into maintenance mode, move the files to their required directories, install the custom object, update permissions, bring the application out of maintenance mode, and restart dynamo.

11. Remove the temporary directory, after the custom object has been installed, as the files are no longer needed:

```
rm -rf darktrace
```

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	>=1.4.1	N/A
python-dateutil	N/A	N/A

Installation

The following provides you with steps on installing a Python 3 Virtual Environment and installing the connector.

> ⚠ The connector requires that you install the Asset on your ThreatQ instance if your are on ThreatQ version 5.9.0 or earlier. The object must be installed prior to installing the connector. See the Asset Object entry under the Prerequisites chapter for more details.

Creating a Python 3.6 Virtual Environment

Run the following commands to create the virtual environment:

```
mkdir /opt/tqvenv/
sudo yum install -y python36 python36-libs python36-devel python36-pip
python3.6 -m venv /opt/tqvenv/<environment_name>
source /opt/tqvenv/<environment_name>/bin/activate
pip install --upgrade pip
pip install threatqsdk threatqcc python-dateutil
pip install setuptools==59.6.0
```

Proceed to Installing the Connector.

Installing the Connector

> ⚠ **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. Navigate to the ThreatQ Marketplace and download the .whl file for the integration.
2. Activate the virtual environment if you haven't already:

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

3. Transfer the whl file to the `/tmp` directory on your ThreatQ instance.
4. Install the connector on your ThreatQ instance:

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

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

5. 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/
```

6. Perform an initial run using the following command:

```
/opt/tqvenv/<environment_name>/bin/tq-conn-darktrace -ll /var/log/tq_labs/ -c /etc/tq_labs/ -v3
```

7. 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.

PARAMETER	DESCRIPTION
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-darktrace -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
```

You will still need to configure and then enable the connector.

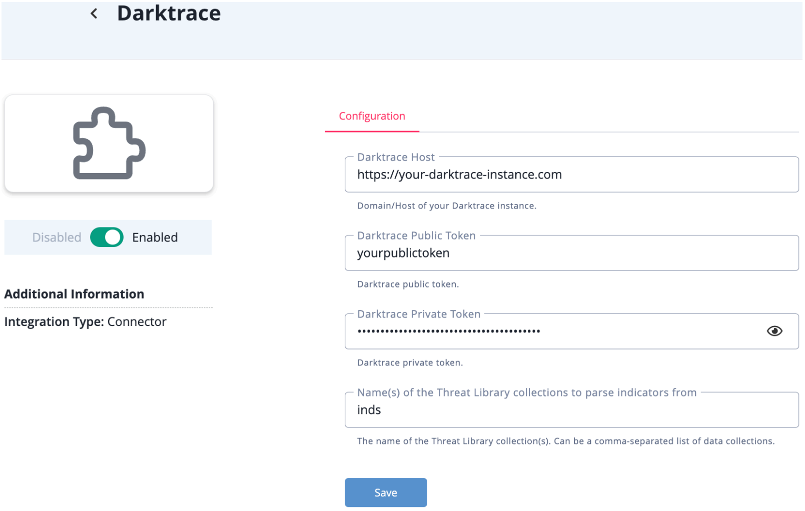
Configuration

> 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
Darktrace Host	The hostname or IP Address of your Darktrace instance.
Darktrace Public Token	The public token for your Darktrace instance.
Darktrace Private Token	The private token for your Darktrace instance.
Name(s) of the Threat Library collections to parse indicators from	Enter a comma-separated list of Threat Library collections from which to export FQDN and IP Address indicators into Darktrace.



Additional Information

Integration Type: Connector

Darktrace Host: https://your-darktrace-instance.com

Domain/Host of your Darktrace instance.

Darktrace Public Token: yourpublictoken

Darktrace public token.

Darktrace Private Token: ••

Darktrace private token.

Name(s) of the Threat Library collections to parse indicators from: inds

The name of the Threat Library collection(s). Can be a comma-separated list of data collections.

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.

Usage

Use the following command to execute the driver:

```
/opt/tqvenv/<environment_name>/bin/tq-conn-darktrace -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.
`-ep, --external-proxy`	Optional - This enables a proxy to be used to contac the internet for the data required by this connector. This specifies an internet facing proxy, NOT a proxy to the TQ instance.
`-d, --no-differential`	Optional - If exports are used in this connector, this will turn 'off' the differential flag for the execution. This allows debugging and testing

ARGUMENT	DESCRIPTION
	to be done on export endpoints without having to rebuild the exports after the test. THIS SHOULD NEVER BE USED IN PRODUCTION.
`-ds, --disable-ssl`	Optional - Disable SSL verification.
`-hist HISTORICAL, --HISTORICAL`	Optional - The number of past days of Darktrace data to load. The default setting is 5.

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/tq-conn-darktrace -c /etc/tq_labs/ -ll /var/log/tq_labs/ -v3
   ```

4. Save and exit CRON.

Change Log

- **Version 1.0.0 rev-a (Guide Update)**
    - Updated the Prerequisites chapter regarding the Asset object. ThreatQ version 5.10.0 introduced the Asset object as a seeded default system object. Users on ThreatQ 5.10.0 or later do not have to install the Asset custom object prior to installing the integration.
- **Version 1.0.0**
    - Initial release