ThreatQuotient

ThreatQ Connector for Microsoft Azure Sentinel

Version 1.5.1

March 15, 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.

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	1.5.1
Compatible with ThreatQ Versions	>= 4.56.0
Python Version	3.6
Support Tier	ThreatQ Supported

Introduction

The ThreatQ Connector for Microsoft Azure Sentinel integration allows a user to export indicators directly to Microsoft Sentinel.

> You must configure a new application in Microsoft Azure before you can install the connector. See the Prerequisites chapter before attempting to install the connector.

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
```

Permissions

The connector requires the `ThreatIndicators.ReadWrite.OwnedBy` permission to be enabled for the ThreatQ Integration App, as both a **delegated right** as well as an **application right**.

Configure New Application

Before installing the integration on the ThreatQ side, you will need to configure a new application on Microsoft Azure. The following link will take you to Microsoft's documentation on how to connect Azure Sentinel to ThreatQ via an Azure Application. In the guide, you can skip step 4 as that step is handled by the ThreatQ integration.

https://docs.microsoft.com/en-us/azure/sentinel/connect-threat-intelligence#connect-azure-sentinel-to-your-threat-intelligence-platform

Alternatively, you can follow the instructions from the link below, without skipping any of the steps:

https://learn.microsoft.com/en-us/azure/sentinel/connect-threat-intelligence-tip#connect-azure-sentinel-to-your-threat-intelligence-platform

Use the following steps if you select **Azure Sentinel** as the **Target** in the connector configuration settings.

https://learn.microsoft.com/en-us/azure/sentinel/connect-threat-intelligence-upload-api

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
requests	N/A	N/A
threatqsdk	>= 1.8.7	N/A
threatqcc	>= 1.4.2	N/A
python-dateutil	N/A	N/A
six	N/A	N/A

Installation

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

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 setuptools==59.6.0 threatqsdk threatqcc python-dateutil six
```

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_ms_sentinel-<version>-py3-none-any.whl
```

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

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-ms-sentinel -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.
ThreatQ Client ID	This is the OAuth id that can be found at Settings Gear → User Management → API details within the user's details.

PARAMETER	DESCRIPTION
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.

Example Output

```
/opt/tqvenv/<environment_name>/bin/tq-conn-ms-sentinel -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>
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
Tenant IDs	Your Microsoft Active Directory App's Tenant ID.
Client ID	Your Microsoft Active Directory App's Client ID.
Client Secret	Your Microsoft Active Directory App's Client Secret.
Saved Search Name (Threat Library Data Collection)	The Threat Library data collection that you want IOCs to be exported from.
Target Product	The target product where IOCs are sent to. Options Include: ◦ Azure Sentinel (default) ◦ Microsoft Defense ATP
Workspace ID	The ID of the Azure Workspace where IOCs are sent to. It should be populated only if target product is **Azure Sentinel**.
Action	The action to take when an IOC is observed in your environment. Options Include: ◦ Unknown ◦ Allow

	◦ Block ◦ Alert
Default Severity	The default severity, between 0 and 5, to apply to the exported IOCs. This can be overridden by attributes. See the Default Values Override section in the Usage Chapter.
Default Threat Type	The default threat type to apply to the exported IOCs. This can be overridden by attributes. See the Default Values Override section in the Usage Chapter. Options Include: ◦ Botnet ◦ C2 ◦ CryptoMining ◦ Darknet ◦ DDoS ◦ MaliciousUrl ◦ Malware ◦ Phishing ◦ Proxy ◦ PUA ◦ WatchList (default)
Default Expiration	The default expiration for exported IOCs. This is used when an indicator does not have an expiration. Options Include: ◦ 2 Weeks (default) ◦ 1 Month ◦ 3 Months ◦ 6 Months ◦ 1 Year ◦ 5 Years
ThreatQ Host / IP Address	The Hostname or IP for your ThreatQ instance. This is so you can link directly back to ThreatQ from Azure.
Behaviour for URL indicators without a scheme defined	Defines how data collection URL indicators should be handled. Options include: ◦ Skip Indicators ◦ http ◦ https

Disable Multiple Runs	Enabling this option disables the PID lock functionality to allow multiple instances to run at once.
Kill Previous Runs After x Seconds	Override the default PID lock timeout. The default timeout is 21,600 seconds (6 hours).
ThreatQ Data Extract Timeout	Set how many seconds to wait for ThreatQ API to return the IOCs that will be exported. The default timeout setting is 600 seconds. If you leave this field blank, the connector to wait indefinitely.

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-ms-sentinel -v3 -c /etc/tq_labs/
-ll /var/log/tq_labs/
```

Default Values Override

The default values configured in the UI for `Threat Type` and `Severity` can be overwritten by indicator attributes as follows:

- The `Default Severity` value can be overwritten by adding an indicator attribute whose name is `Severity` and whose value is between 0-5 (inclusive).
- The `Default Threat Type` value can be overwritten by adding an indicator attribute whose value is a valid `Threat Type` value (based on the above options for `Default Threat Type`) or is an alias of a valid `Threat Type` based on the following mapping:

```
aliases = {
    'C2': ['command and control', 'c&c', 'command & control'],
    'DDoS': ['denial of service'],
    'CryptoMining': ['crypto', 'mining', 'crypto miner'],
    'Botnet': ['bot']
}
```

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 Intelligence Mailbox connector instances on the same TQ box).
`-hist, --historical {DATE}`	Optional - Allows you to set the start date for the Threat Library search.
`-ep, --external-proxy`	Optional - Enables a proxy to be used to contact the internet for the data required.

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-ms-sentinel -
   c /etc/tq_labs/ -ll /var/log/tq_labs/ -v3
   ```
4. Save and exit CRON.

Known Issues / Limitations

- The Microsoft Graph API has a hard time handling more than 100 IOCs within an upload at one time. The API will throw a gateway error, saying the upload timed-out. Any upload errors will be retried.
- The Microsoft Graph API will automatically de-duplicate and update IOCs that are sent to their API.
- The Microsoft Azure Upload Indicators API requires a STIX identifier for each indicator updated to Microsoft Sentinel. As of this publication, the identifier is randomly generated.

Change Log

- **Version 1.5.1**
  - Added new configuration field: **ThreatQ Data Extraction Timeout**.
- **Version 1.5.0**
  - The integration will now automatically re-authenticate with Microsoft Azure Sentinel when it receives a 401/403 status code.
- **Version 1.4.6**
  - Added PID Lock Support for the integration.
  - Added two new configuration fields:
    - Disable Multiple Runs
    - Kill Previous Runs After x Seconds
  - Updated the Known Issues / Limitations chapter.
- **Version 1.4.5 rev-a**
  - Guide Update - updated links in the Configure New Application section under the Prerequisites chapter.
- **Version 1.4.5**
  - Added new configuration option, **Workspace ID**, which allows you to enter the ID of the Azure Workspace where IOCs are sent to.
  - Replaced Microsoft Graph API with Microsoft Azure Upload Indicators API when the Azure Sentinel target product is selected.
  - Malware Family and Adversary data are now sent in the description for the Azure Sentinel target product.
  - Removed support for uploading indicators of type ASN.
  - Updated ThreatQ SDK integration dependency to 1.8.7.
- **Version 1.4.4**
  - Added a new configuration option, **Behaviour for URL indicators without a scheme defined**, to help resolve `206 - Partial Content` errors.
- **Version 1.4.3**
  - Added proxy support for the integration.
- **Version 1.4.2**
  - Fixed an issue where objects without descriptions would cause an `uploading indicators: a bytes-like object is required` error when running the connector.
- **Version 1.4.1 rev-a**
  - Guide Update - Updated pathways in virtual environment steps and connector commands.
- **Version 1.4.1**
  - Fixed an error that occurred when uploading indicators that included a hard space.
- **Version 1.4.0**
  - Fixed an type object argument error that would cause the connector to fail with an error of `Unable to connect the Microsoft Sentinel connector with ThreatQ: type object argument after ** must be a mapping, not unicode`.
- **Version 1.3.1**

  - Fixed a backward compatibility issue.
- **Version 1.3.0**
  - Updated request parameters for IP Addresses.
  - Added additional debug logs.
- **Version 1.2.0**
  - Fixed a user field bug.
  - Added additional debug logs.
  - Added Python 3 support.
- **Version 1.1.0**
  - Connector now includes Malware Family from attributes of indicators contained in the Threat Library saved search.
  - Connector now includes Adversary data from indicator relationships.
  - Added the ability to sanitize and strip HTML characters from indicator descriptions.
- **Version 1.0.0**
  - Initial Release