ThreatQuotient

Microsoft Azure Sentinel Connector Guide

Version 1.1.0

June 16, 2021

ThreatQuotient
11400 Commerce Park Dr., Suite 200
Reston, VA 20191

Support
Email: support@threatq.com
Web: support.threatq.com
Phone: 703.574.9893

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.

Contents

Versioning

- Current integration version: `1.1.0`
- Supported on ThreatQ versions >= `4.38.0`

Introduction

The Microsoft Azure Sentinel Connector for ThreatQ integration allows a user to export indicators directly to Microsoft Sentinel via Microsoft's Graph API.

Prerequisites

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

Installation

The connector can be installed from the ThreatQuotient repository with YUM credentials or offline via a .whl file.

> **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. Install the connector using one of the following methods:

   ThreatQ Repository

   a. Run the following command:

   ```
   <> pip install tq_conn_ms_sentinel
   ```

   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:

   ```
   <> mkdir /tmp/ tq_conn_ms_sentinel

      pip download tq_conn_ms_sentinel -d

      /tmp/ tq_conn_ms_sentinel/
   ```

   b. Archive the folder with the .whl files:

   ```
   <> tar -czvf tq_conn_ms_sentinel.tgz /tmp/
      tq_conn_ms_sentinel/
   ```

   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_conn_ms_sentinel.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.

```
<> pip install /tmp/conn/ tq_conn_ms_sentinel-<version>-py2-
   none-any.whl --no-index --find-links /tmp/conn/
```

> A driver called `tq-conn-ms-sentinel` will be installed. After installing with `pip` or `setup.py`, a script stub will appear in `/usr/bin/tq-conn-ms-sentinel`.

2. 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
```

3. Perform an initial run using the following command:

```
<> tq-conn-ms-sentinel -v3 -ll /var/log/tq_labs/ -c /etc/tq_labs/
```

4. Enter the following parameters when prompted:

PARAMETER	DESCRIPTION
File Log Location	This is the pathway to your log location.
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 User in the ThreatQ System for integrations.

PARAMETER	DESCRIPTION
ThreatQ Password	The password for the above ThreatQ account.

Example Output

```
tq-conn-ms-sentinel -v3 -ll /var/log/tq_labs/ -c /etc/tq_labs/
File Log Location: <Pathway to log>
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 to open its details page.
4. Enter the following parameters under the **Configuration** tab:

PARAMETER	DESCRIPTION
Tentant 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

PARAMETER	DESCRIPTION
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.
Defaut Threat Type	The default threat type to apply to the exported IOCs. This can be overridden by attributes. Options Include: • Botnet • C2 • CryptoMining • Darknet • DDoS • MaliciousUrl • Malware • Phishing • Proxy • PUA • WatchList (default)

PARAMETER	DESCRIPTION
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.

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.

Usage

Use the following command to execute the driver:

```
<> tq-conn-ms-sentinel -v3 -ll /var/log/tq_labs/ -c /etc/tq_labs/
```

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

- 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`	Shows this help message and exits.
`-n NAME, --name NAME`	This sets the name for this connector. In some cases, it is useful to have multiple connectors of the same type executing against a single TQ instance. For example, the Syslog Exporter can be run against multiple target and multiple exports, each with their own name and configuration
`-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. The default setting is **1** (Warning).

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 * * * 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.

Change Log

- **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