ThreatQuotient

Proofpoint TAP Connector Guide

Version 1.3.1

April 15, 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: `1.3.1`
- Supported on ThreatQ versions >= `4.30.0`

There are two versions of this integration:

- Python 2 version
- Python 3 version

Introduction

Proofpoint TAP is a service that analyzes, detects and helps mitigate attacks that target people via email. The analysis data for each email that has been flagged as malicious by TAP is available via their API. The API provides multiple endpoints, of which we use the following:

SIEM

The SIEM endpoint allows integration with these solutions by giving administrators the ability to periodically download detailed information about several types of TAP events in a SIEM-compatible, vendor-neutral format. Currently, the following event types are exposed:

- Blocked or permitted clicks to threats recognized by URL Defense
- Blocked or delivered messages that contain threats recognized by URL Defense or Attachment Defense

Campaign

The Campaign endpoint allows administrators to pull specific details about campaigns, including:

- Their description
- The actor, malware family, and techniques associated with the campaign
- The threat variants which have been associated with the campaign

Forensics

The Forensics endpoint allows administrators to pull detailed forensic evidences about individual threats or campaigns observed in their environment. These evidences could be used as indicators of compromise to confirm infection on a host, as supplementary data to enrich and correlate against other security intelligence sources, or to orchestrate updates to security endpoints to prevent exposure and infection.

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_proofpoint_tap
   ```

   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_proofpoint_tap

      pip download tq_conn_proofpoint_tap -d

      /tmp/ tq_conn_proofpoint_tap/
   ```

   b. Archive the folder with the .whl files:

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

   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_proofpoint_tap.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_proofpoint_tap_<version>-
   <python version>-none-any.whl --no-index --find-links /
   tmp/conn/
```

EX pip install /tmp/conn/ tq_conn_proofpoint_tap-1.3.1-py2-none-any.whl --no-index --find-links /tmp/conn/

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

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-proofpoint-tap -v3 -ll /var/log/tq_labs/ -c /etc/
   tq_labs/
```

4. 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 at Settings Gear → User Management → API details within the user's details.
Email Address	This is the User in the ThreatQ System for integrations.
Password	The password for the above ThreatQ account.

PARAMETER	DESCRIPTION
Status	This is the default status for objects that are created by this Integration.

Example Output

```
tq-conn-proofpoint-tap -v3 -ll /var/log/tq_labs/ -c /etc/tq_labs/
ThreatQ Host: <ThreatQ Host IP or Hostname>
Client ID: <ClientID>
E-Mail Address: <EMAIL ADDRESS>
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 to open its details page.
4. Enter the following parameter under the **Configuration** tab:

PARAMETER	DESCRIPTION
Hostname	Hostname or IP address of the Proofpoint TAP API.
API Principal	The principal provided for the API.
API Secret	The secret provided for the API.
Number of Hours to Pull Data from History	An integer representing a historical time window in hours to pull data from the SIEM all endpoint. The max historical timeframe is 24 hours.
Do you want to create an Incident in ThreatQ if Proofpoint TAP Reports tha a User has clicked on a URL?	Proofpoint TAP returns data if it has detected that a user has clicked on a malicious URL. If this flag is set to Yes, the integration will automatically create incidents in ThreatQ.

5. Review the **Settings** configuration, 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-proofpoint-tap -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`	Shows this help message and exits.
`-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).
`-ep, --external-proxy`	This allows you to use the proxy that is specified in the ThreatQ UI.
`-ds, --disable-ssl`	Adding this flag will disable SSL verification when contacting the Metron API.

ARGUMENT	DESCRIPTION
`-dp, --disable-proxy`	This flag will allow the connector to bypass the proxy when running.

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-proofpoint-tap -c /etc/tq_labs/ -ll /var/
      log/tq_labs/ -v3
   ```

4. Save and exit CRON.

Change Log

- **Version 1.3.1**
  - Adeded a flag, -dp, that allows the connector to bypass proxy when running.
  - Added new configration parameter for creating incidents.
- **Version 1.3.0**
  - Added Python 3 support.
- **Version 1.2.0**
  - Added the ability to open incident response tickets if the user has requested it via the ThreatQ UI.
- **Version 1.1.0**
  - Improved exception handling
  - Optimized API calls to Proofpoint TAP
- **Version 1.0.0**
  - Initial Release