McAfee TIE Connector Implementation Guide

Version 1.2.0

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.

Last Updated: Wednesday, October 3, 2018

Contents

Introduction

This connector (integration) will interact with the McAfee TIE server. The TIE server is a database of malicious files and their reputations. The integration will pull the indicator hashes from the ThreatQ Threat Library, perform a potentially custom mapping of indicator attributes to the McAfee file reputations, and then push these indicators to the TIE server.

Installation

This integration is installed via `pip`. First, modify the `pip.conf` file on your environment as shown below:

```
[global]
   index-url = https://system-updates.threatq.com/pypi
   extra-index-url = https://<username>:<password>@extensions.threatq.com/
                               threatq/integrations
                       https://<username>:<password>@extensions.threatq.com/threatq/sdk
```

The following command installs this integration. The current version is `1.2.0`.

```
pip install tq_conn_mcafee_tie
```

Current Features

- User configurable rate limiting: ThreatQ will not push more than the configured indicators per day in the TIE server. *1000 indicators per day* is the hard limit. The rate limit is honored regardless of how often the connector runs.
- ThreatQ indicator scores are mapped to McAfee reputation scores via user configuration.

- A user can export only indicators of interest out of the ThreatQ platform via configuration.
- Ability to use McAfee EPO's provisioning capability to get a signed certificate for communication with the TIE server.
- Ability to enrich any hash indicators in TQ sent to McAfee TIE with any additional information from the McAfee ecosystem.
- **New in Version 1.2.0**: Ability to enrich any hash indicators in ThreatQ sent to McAfee TIE with any additional information from the McAfee ecosystem.

Usage

Provisioning

To communicate with the McAfee TIE server, the user needs to have a certificate and the equivalent Certificate Authority (CA) needs to be imported in the McAfee EPO. The steps are documented here: https://opendxl.github.io/opendxl-client-python/py-doc/epoexternalcertissuance.html.

However, there is an easier alternate way that uses a McAfee command line provisioning tool. **For simplicity of use, this approach is recommended**. This integration wraps the McAfee commandline provisioning tool and provides a commandline utility that can be invoked as follows:

```
tq-mcafee-tie-prov --epo-ip <epo_ip> --epo-login <epo_login> --epo-pass <epo_pass>
```

You can also pass a nonstandard EPO port and other optional arguments to the program above. To find additional options, simply invoke the program with `-h`.

If it is undesirable to supply the password on the command line, you can omit it and instead invoke the utility as:

```
tq-mcafee-tie-prov --epo-ip <epo_ip> --epo-login <epo_login>
```

The program will prompt you for the password.

If this connector is being used to connect to multiple DXL brokers, the `-cn` or `--conn-name` option will be used to differentiate among those connectors. This option should be passed to the provisioning script (and the exact same `--conn-name` should be used in the actual connector as described below).

Connector

The connector cannot be invoked until a directory containing the DXL configuration file is created either by the above provisioning commandline utility or by manually following the steps provided by the OpenDXL documentation.

The connector command line utility can be invoked as follows (all command line options are optional):

```
tq-mcafee-tie -ll <log_location_or_stdout> -c <tq_config_location> -x <mcafee_tie_cache_file_location> -dc <dxl_cert_dir_location> -cn <connector-name>
```

All location-based options default to the current working directory if they are not provided. To find additional options and option descriptions, simply invoke the program with `-h`.

The connector requires some setup before it can start communicating with ThreatQ and the McAfee DXL server:

1. Run the `tq-mcafee-tie` commandline utility in order for it to create a ThreatQ configuration file and register itself to the ThreatQ server for you.
2. Login to the ThreatQ user interface.

3. Navigate to **Incoming Feeds > Labs**.
4. Click **Feed Settings** for the McAfee TIE connector.
5. Click the toggle button to the left of the McAfee TIE connector name. The connector name in the user interface will be `McAfee TIE-<connector-name>` if the user passed a `-cn` option on the commandline.

After the above is setup correctly, the next invocation of the `tq-mcafee-tie` (or `tq-mcafee-tie --cn <connector-name>`) command line utility will grab file hash indicators from ThreatQ based on the Daily Rate Limiting value in order to set their reputation level in the configured McAfee TIE server.

> If you have registered multiple connectors, you must ensure you are passing the appropriate connector name suffix via the `-cn` option when invoking the CLI command.

Connector Configuration

1. This connector gives the users the ability to specify a daily limit for the number of DXL set reputation requests made. By default, this limit is 1000. Valid values for the Daily Rate Limiting are 1-1000. The daily rate limit helps to prevent overloading the McAfee TIE infrastructure.

   > Data pertaining to the daily rate limiting is persisted in the file `mcafee_tie_cache.json` (or `mcafee_tie_cache-<connector-name>.json`). This file can be modified or removed to reset the daily rate limit for the specific connector in question.

2. TQ scoring to McAfee TIE reputation mapping: This connector gives the users the ability to map ThreatQ scoring bands to specific McAfee TIE reputation values.

   ThreatQ supports the following scoring bands:

ThreatQ Scoring Band	Range
very low	0-4
low	5-6
medium	7-8
high	9
very high	10+

One or more ThreatQ scoring band can be mapped to a McAfee reputation score with the following conditions:

- The same ThreatQ scoring band cannot be mapped to multiple McAfee TIE reputation scores.
- A higher ThreatQ scoring band cannot be mapped to a less malicious McAfee TIE reputation score. For example, the following configuration is **invalid**:

McAfee TIE Reputation	ThreatQ Scoring Bands
Known Malicious	low
Likely Malicious	medium
Maybe Malicious	high

- Multiple scoring bands can be assigned to the same Reputation as long as the above two conditions are satisfied. An example of a valid configuration is as follows:

McAfee TIE Reputation	ThreatQ Scoring Bands
Known Malicious	very high, high
Likely Malicious	medium
Maybe Malicious	low, very low

3. This connector gives the users the ability to filter on which indicators to send to the McAfee TIE server.

   The following McAfee TIE connector user fields can be used for indicator filtering. All fields are optional.

Field	Description	Valid Values	Default Value
Number of Days	Filters by indicators added in the last N days	1-365	N/A
Filter by Indicator Status	Filter by indicators that have the provided indicator status name. Accepts only a single status name.	Indicator status names returned by GET /ap-i/indicator/statuses	Active
Filter by Indicator Score	Filters by indicators that have a score ≥ the provided score. An indicator's manual score	0-10	N/A

Field	Description	Valid Values	Default Value
	has greater precedence than its generated score. A provided score of 0 is effectively the same as if this filter weren't applied at all.		
Filter by Indicator Attributes	A comma-separated list of key-value pairs in which the key corresponds to the Attribute Name and the value corresponds to the Attribute Value. The key and value are separated by a colon. The user can filter on multiple Attribute Values for the same Attribute Name by having individual key-value pairs where the key is the same for both pairs. Example: `Attribute1:Value1, Attribute2:Value2, Attribute1:Value2`	N/A	N/A

4. The users can specify whether they want to automatically enrich indicators sent to McAfee with additional information from McAfee ecosystem.

   The following attributes are created for each of McAfee ATD, McAfee GTI and the Enterprise (which in this case is ThreatQ).

Attribute	Possible Values	Description
Reputation	See Appendix 4	Reputation of this hash as known by either ATD, GTI or Enterprise
Prevalence	Integer > 0	The prevalence data for this hash
Created At	Valid Date	The time this hash was first seen
Count	Integer > 0	Endpoints on which this hash was seen

Appendix

`mcafee_tie_cache.json` - Cache utilized by the connector. The following key value pairs are recorded in the cache file:

1. `indicators_sent`: The total number of indicators sent to McAfee TIE over a 24 hour period.
2. `start_time`: The epoch time from when the connector was initially ran. This value is set to the current time whenever the connector runs 24 hours after the recorded start time.
3. `last_recieved_indicator_id`: The id of the last indicator sent to McAfee TIE.
4. The values for `ATD Trust Level` attribute can be as follows:
    - Reputation 1 --> Known Malicious
    - Reputation 15 --> Mostly Malicious
    - Reputation 30 --> Likely Malicious
    - Reputation 50 --> Unknown
    - Reputation 70 --> Likely Trusted
    - Reputation 85 --> Mostly Trusted
    - Reputation 100 --> Known Trusted
    - Reputation 0 --> Not Set

Known Issues

On MAC High Sierra, an existing bug that doesn't support libressl can cause the EPO provisioning tool `tq-mcafee-tie-prov` to fail with a message like: `Error initializing ctypes`. You can manually apply a patch as documented here: https://github.com/wbond/oscrypto/commit/83e2a06085b5b09ee5cd6c28a2ae1c52352c9e53