FireEye CMS Getting Started Guide

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

Last Updated: Sunday, March 25, 2018

Introduction

This connector is unidirectional, pulling alerts from FireEye CMS and uploading the data as indicators and events to a ThreatQ instance. The events are tagged as "Malware" type events. It attaches to a single FireEye CMS instance. The `-n` flag may be used for multiple instances, but it is not a tested use case.

Installation

This package is available from the threatq pypi extensions index.

To install, run the following command:

```
# pip install -i https://<USERNAME>:<PASSWORD>@ex-
tensions.threatq.com/threatq/integrations tqFireEye
```

Where *USERNAME* and *PASSWORD* are the username and password provided by ThreatQ to enable updates to the appliance.

To upgrade, run the following command:

```
# pip install --upgrade https://<USERNAME>:<PASSWORD>@ex-
tensions.threatq.com/threatq/integrations tqFireEye
```

Executing the Driver

This package comes with a driver called `tq-fireeye`. After installing with `pip`, a script stub will appear in `/bin/tq-fireeye`.

First, create a directory, which will house any downloaded files.

To execute the feed, simply run:

```
#> tq-fireeye -c /path/to/config/directory/ -ll /path/to/-
log/directory/ -ds -v 3
```

OR

```
#> tq-fireeye -c /path/to/config/directory/ -ll /path/to/-
log/directory/ -ds -v3
```

The `-ds` flag is used to disable SSL verification. This should be used if the connected FireEye CMS instance uses a self-signed certificate. By default, FireEye CMS uses a self-signed certificate.

To use the proxy settings specified in your ThreatQ user interface, append the `-ep` flag to the end.

If you want to parse data from a JSON file (instead of the API), append the `-f /path/to/your/file` flag to the end of your command. This will parse the file, but not change the internal time record for when this connector last connected to FireEye CMS. Ingesting duplicate data will not cause issues, as the data is recorded the same way.

The driver will run once, where it will connect to the TQ instance and install the user interface component of the connector.

Running it subsequent times will cause an ingestion of FireEye data alert.

CRON

To run this script on a recurring basis, use CRON or some other on system schedule. The argument in the cron script must specify the config and log locations.

This can be run multiple times a day and should not be run more often than once every few minutes.

Driver command line options

The FireEye driver has several command line arguments that will help you and your customers execute it. They are listed below. You can view the commands by executing `tq-fireeye -h` or `tq-fireeye -h`.

```
usage: FireEye Connector [-h] [-ll LOGLOCATION]
                         [-c CONFIG] [-v VERBOSITY]

This is the Fireeye Connector

optional arguments:
       -h, --help      show this help message and exit
       -ll LOGLOCATION, --loglocation LOGLOCATION
                       This sets the logging location for
                       this connector. The location should
                       exist and be writable by the current
                       user. A special value of 'stdout' means
                       to log to the console (this happens by
                       default)
       -c CONFIG, --config CONFIG
```

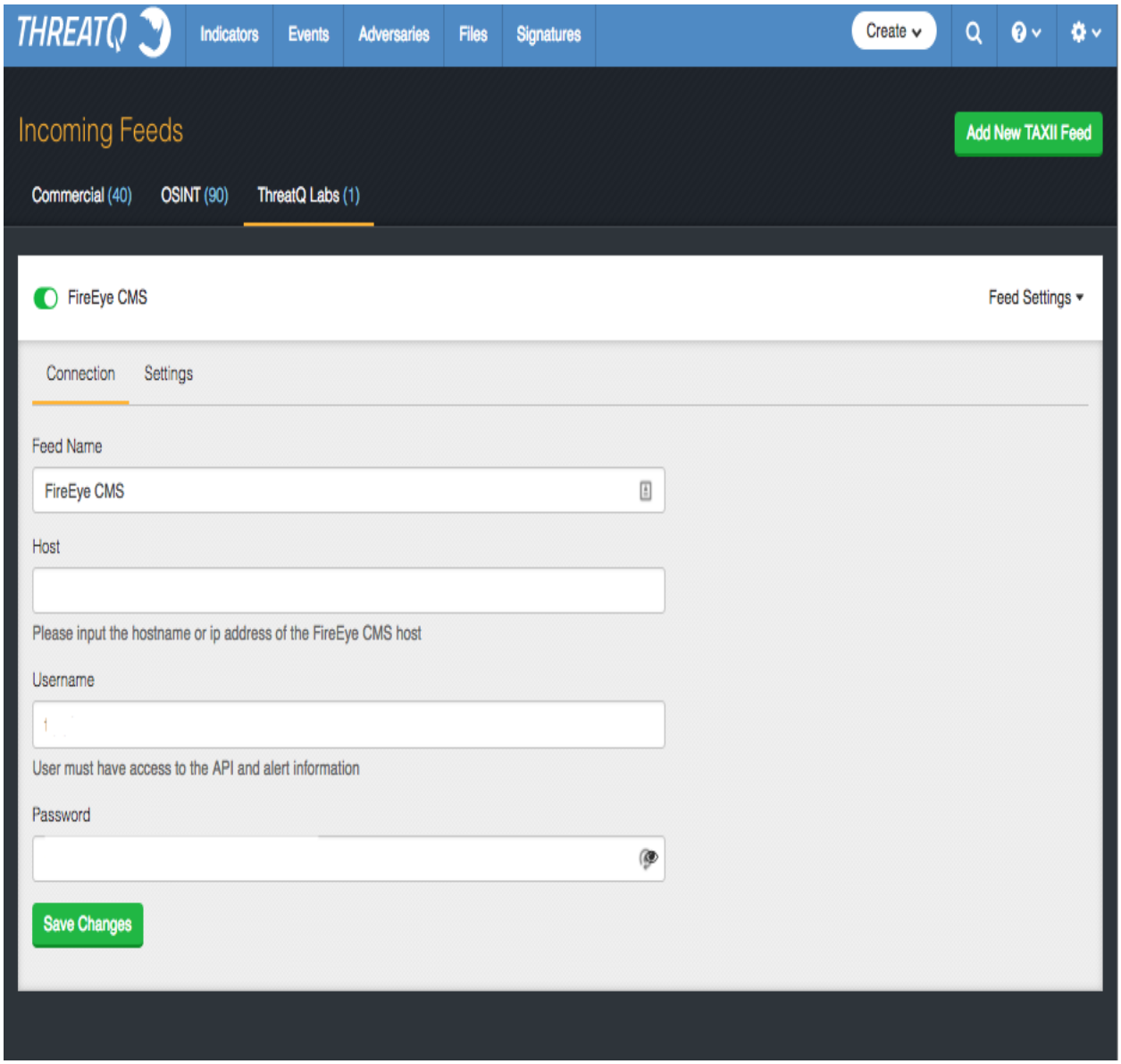
```
                    This is the location of the configuration
                    file for the connector. This location must
                    be readable and writable by the current user.
                    If no config file is given, the current
                    directory will be used. This file is also
                    where some information from each run of
                    the connector may be put (e.g. last run time,
                    private oauth, etc)
  -v {1,2,3}, --verbosity {1,2,3}
                    This is the logging verbosity level. The
                    Default is 1 (Warning)
  -f, --file        Allows you to specify parsing JSON of
                    a file instead of using the API endpoint
                    of FireEye CMS. Used for offline
                    data pulls.
  -ds, --disable_ssl    This allows you to disable SSL
                        verification to all
                        requests to the FireEye CMS API.
  -ep, -external-proxy  This allows you to use the proxy
                        that is specified in the ThreatQ UI.
```

Most commonly, if you want to see the output on the command line, execute the following command:

```
tq-fireeye -ll stdout -v 3 or tq-fireeye -ll stdout -v 3
```

Configuration

Once installed, you will need to configure the connector from the ThreatQ user interface. To complete this task, navigate to the **Incoming Feeds** page in ThreatQ and click on the **ThreatQ Labs** tab. Find the entry for **FireEye** and switch the integration on. Then enter configuration options.



This information will be used to login to FireEye CMS to pull in the alerts to create indicators/events in ThreatQ.

- **Host:** Is the hostname or the IP Address of the FireEye CMS instance
- **Username/Password:** Are the credentials of a FireEye CMS user that can use the API and has access to Alerts