ThreatQuotient

Events Builder Connector Guide

Version 1.0.3 rev-a

March 23, 2022

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

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.

Versioning

- Current integration version: `1.0.2`
- Compatible with ThreatQ versions >= `4.34.0`
- Python version: `3.6`

Introduction

The Events Builder Connector accepts a provided payload, parses it, and creates an event object in ThreatQ with attributes and related indicators. The events ingested into ThreatQ can include related objects such as Sources, Targets, Identities, and Assets.

The payload is defined in the command used to execute the driver. See the Usage chapter for more information regarding the payload flag.

You can identify your private IP Addresses in the connector's UI configuration page in ThreatQ to prevent them from being added as Indicators.

Prerequisites

Review and confirm that you have fulfilled the connectors prerequisites before proceeding with installation/upgrading.

Files and Directories

Throughout this guide there will be referrals to several files and directories, some which will be symbolic, while others may change depending on the specifics of the environmental setup.

Timezone

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
```

PIP.conf

Prior to ThreatQ version 4.10, you were required to modify your system's pip.conf to use the ThreatQ integrations python repo, also known as DevPi. This functionality was made available upon an initial install of 4.10. If you have upgraded to 4.10 from a previous version, you will need to modify the pip.conf on your environment to the following (replacing username and password with your information).

```
[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
```

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
threatqsdk	>= 1.8.2	N/A
threatqcc	>= 1.4.1	N/A
ipaddress	N/A	N/A
python-dateutil	N/A	N/A
ruamel.yaml	**0.14.11**	**Pinned Version**

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
pip install threatqsdk threatqcc
pip install python-dateutil
```

Proceed to installing the connector.

Installing the Connector

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. Activate the virtual environment:

   ```
   source /opt/tqvenv/<environment_name>/bin/activate
   ```

   b. Run the following command:

   ```
   pip install tq_conn_events_builder
   ```

   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_events_builder

   pip download tq_conn_events_builder -d /tmp/
   tq_conn_events_builder/
   ```

   b. Archive the folder with the .whl files:

   ```
   tar -czvf tq_conn_events_builder.tgz /tmp/
   tq_conn_events_builder/
   ```

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

> A driver called `tq-conn-events-builder` will be installed. After installing with `pip` or `setup.py`, a script stub will appear in `/opt/tqvenv/<environment_name>/bin/tq-conn-events-builder`.

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:

```
/opt/tqvenv/<environment_name>/bin/tq-conn-events-builder
-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.

PARAMETER	DESCRIPTION
Password	The password for the above ThreatQ account.
Status	This is the default status for objects that are created by this Integration. This value is case sensitive. Example: Active.

Example Output

```
/opt/tqvenv/<environment_name>/bin/tq-conn-events-builder -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:

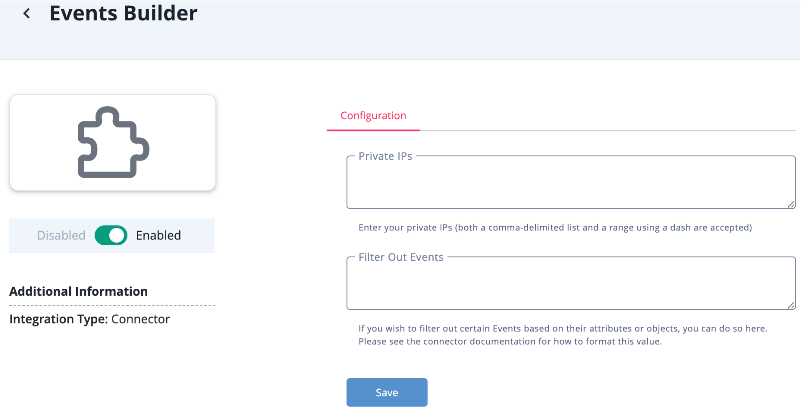
1. Navigate to your integrations management page in ThreatQ.
2. Select the **Labs** option from the *Category* dropdown (optional).
3. Click on the integration to open its details page.
4. Enter the following parameter under the **Configuration** tab:

<table>
<tr><th>PARAMETER</th><th>DESCRIPTION</th></tr>
<tr><td><b>Private IPs</b></td><td>Enter your private IPs so that IP Addresses will not be added as Indicators if they are private. Both, a comma-delimited list and a range using a dash, formats are accepted.</td></tr>
<tr><td><b>Filter Out Events</b></td><td>You can filter out certain events based on their attribute names. This value should be in following format:
<pre>

  <Syslog source 1>:
    <tq_object>:
        <Attribute Name>: <Attribute Value>
        <Attribute Name>: <Attribute Value>
 <Syslog source 2>:
    <tq_object>:
        <Attribute Name>: <Attribute Value>
</pre>
In the example above, the <code><tq_object></code> is one of the following - attributes, adversaries, asset, attack_pattern, campaign, corporate_email, identity, incident, indicators, malware, uri_source, uri_target, or vulnerability. The <code><Attribute Name></code> would either be exactly an attribute name, a TQ object type (as the URI type) - ie. Identity - or an Indicator Type, ie. IP Address; <code><Attribute Value></code> would simply be the value. If a TQ object has no type, <code><Attribute</code></td></tr>
</table>

<table>
<tr><th>PARAMETER</th><th>DESCRIPTION</th></tr>
<tr><td></td><td>
<p><code>Name>: <Attribute Value></code> would simply be replaced by the value - see malware below below.</p>
<p>If you wish to pass <code>Event Time</code> as a filtered attribute, make sure to format the date as a datetime as demonstrated below.</p>
<p>Additionally, the values can be comma-delimited as seen in the example below.</p>
<p>Example with Sources:</p>
<pre>

  McAfee EPO:
    uri_target:
        Identity: SYSTEM
    attributes:
        Source Process: java.exe
    malware:
        RAT
  McAfee EPO:
    attributes:
        Event Time: 2020-10-01 12:57:18
        Workstation Name: 123AEP, 436asd, sdfs345
</pre>
</td></tr>
</table>



Integration Type: Connector

Private IPs

Enter your private IPs (both a comma-delimited list and a range using a dash are accepted)

Filter Out Events

If you wish to filter out certain Events based on their attributes or objects, you can do so here. Please see the connector documentation for how to format this value.

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:

```
/opt/tqvenv/<environment_name>/bin/tq-conn-events-builder -v3
-ll /var/log/tq_labs/ -c /etc/tq_labs/ -p <JSON payload>
```

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).
`-n, --name`	This allows you to change the name of the connector.
`-d, --no-differential`	If exports are used in this connector, this will turn 'off' the differential flag for the execution. This allows debugging and testing to be done on export endpoints without having to rebuild the

ARGUMENT	DESCRIPTION
	exports after the test. THIS SHOULD NEVER BE USED IN PRODUCTION.
`-ep, --external-proxy`	This enables a proxy to be used to connect to the internet for the data required by this connector. This specifies an internet facing proxy, NOT a proxy to the TQ instance.
`-p PAYLOAD, --payload PAYLOAD`	Dictionary with data needed to create events in ThreatQ - event name, IOCs, attributes, etc.

Payload Example

The following is an example of a payload that could be passed with the connector. This example is from NiFi's source: AmazonUserAddDetection.

```
{
    "event_type":"Sighting",
    "malware":null,
    "identity":"user@co.com",
    "uri_target":null,
    "uri_target_attributes":null,
    "event_name":"AWS User Add Detection: CreateUser from 1.2.3.4 to random.random.com",
    "event_source":"AWS User Add Detection",
    "event_attributes":"Event Domain<|random.random.com~|Target Domain<|rando-rando-1~|User-Agent<|
console.random.com",
    "uri_source":"IP Address:1.2.3.4",
    "syslog_message":{
        "eventVersion":"1.08",
        "userIdentity":{
            "type":"AssumedRole",
            "principalId":"ABCDE6FGHI7JKLM:user@co.com",
            "arn":"arn:aws:sts::123456789:test-role/UserName/user@co.com",
            "accountId":"123456789",
            "accessKeyId":"ZYXWVUT",
            "sessionContext":{
                "sessionIssuer":{
                    "type":"Role",
                    "principalId":"ABCDE6FGHI7JKLM",
                    "arn":"arn:aws:iam::123456789:role/UserName",
                    "accountId":"123456789",
                    "userName":"UserName"
                },
                "webIdFederationData":{},
                "attributes":{
                    "creationDate":"2021-10-05T18:32:56Z",
                    "mfaAuthenticated":"false"
                }
            }
        },
        "eventTime":"2021-10-05T19:01:02Z",
        "eventSource":"random.random.com",
        "eventName":"CreateUser",
        "awsRegion":"rando-rando-1",
        "sourceIPAddress":"1.2.3.4",
        "userAgent":"console.random.com",
        "requestParameters":{
            "userName":"this_username",
            "tags":[
                {
                    "key":"ProductName",
                    "value":"this_username"
                },
                {
                    "key":"Owner",
                    "value":"user@co.com"
                },
                {
                    "key":"Team",
                    "value":"My_Team"
                },
                {
                    "key":"BusinessUnit",
                    "value":"IT"
                }
            ]
        },
        "responseElements":{
            "user":{
                "path":"/",
                "userName":"this_username",
                "userId":"AAABBBCCCDDDEEE1223",
                "arn":"arn:aws:iam::123456789:user/this_username",
                "createDate":"Oct 5, 2021 7:01:02 PM",
                "tags":[
                    {
                        "key":"ProductName",
                        "value":"this_username"
                    },
                    {
                        "key":"Owner",
                        "value":"user@co.com"
                    },
                    {
                        "key":"Team",
                        "value":"My_Team"
                    },
                    {
                        "key":"BusinessUnit",
                        "value":"IT"
                    }
                ]
            }
        },
        "requestID":"111222-a333444-ds555-ab667899",
        "eventID":"1a2b3c-456f7g-890j1",
        "readOnly":false,
        "eventType":"AwsApiCall",
        "managementEvent":true,
        "recipientAccountId":"123456789",
        "eventCategory":"Management",
        "sessionCredentialFromConsole":"true"
    },
    "uri_source_attributes":null,
    "event_happened_at":"2021-10-05T19:01:02Z"
}
```

Change Log

- **Version 1.0.3 rev-a**
  - Guide Update - Updated pathways in virtual environment steps and connector commands.
- **Version 1.0.3**
  - Added new configuration parameter, **Filter Out Events**, to filter out events based on certain attributes/objects.
  - Added event_happened_at as "Event Time" event attribute.
  - Added all event related-object attributes to the main event.
  - Fixed a ThreatQ Adversary upload issue.
- **Version 1.0.2**
  - Initial Release