

ThreatQuotient



Events Builder Connector User Guide

Version 1.2.0

December 17, 2023

ThreatQuotient

20130 Lakeview Center Plaza Suite 400
Ashburn, VA 20147

 **ThreatQ Supported**

Support

Email: support@threatq.com

Web: support.threatq.com

Phone: 703.574.9893

Contents

Warning and Disclaimer	3
Support	4
Integration Details.....	5
Introduction	6
Prerequisites	7
Time Zone	7
Integration Dependencies	7
Installation.....	9
Creating a Python 3.6 Virtual Environment	9
Installing the Connector.....	10
Configuration	12
Usage.....	15
Command Line Arguments.....	15
Payload Example.....	17
Change Log	20

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.

Copyright © 2023 ThreatQuotient, Inc.

All rights reserved. This document and the software product it describes are licensed for use under a software license agreement. Reproduction or printing of this document is permitted in accordance with the license agreement.

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.

Integration Details

ThreatQuotient provides the following details for this integration:

Current Integration Version 1.2.0

**Compatible with ThreatQ
Versions** $\geq 4.34.0$

**Compatible with Third-Party
Products** Apache NiFi 1.12.0+

Python Version 3.6

Support Tier ThreatQ Supported

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.

Time Zone

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
```

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	>= 1.8.7	N/A
------------	----------	-----

DEPENDENCY	VERSION	NOTES
threatqcc	>= 1.4.2	N/A
ipaddress	N/A	N/A
python-dateutil	2.8.2	Pinned Version
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 threatqsdk threatqcc setuptools==59.6.0 python-dateutil==2.8.2
```

Proceed to [Installing the Connector](#).

Installing the Connector

⚠ 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. Navigate to the ThreatQ Marketplace and download the .whl file for the integration.
2. Activate the virtual environment if you haven't already:

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

3. Transfer the whl file to the /tmp directory on your ThreatQ instance.
4. Install the connector on your ThreatQ instance:

```
pip install /tmp/tq_conn_events_builder-<version>-py3-none-any.whl
```



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

5. 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/
```

6. Perform an initial run using the following command:

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

7. 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.
ThreatQ Client ID	This is the OAuth id that can be found at Settings Gear → User Management → API details within the user's details.

PARAMETER	DESCRIPTION
ThreatQ Username	This is the Email Address of the user in the ThreatQ System for integrations.
ThreatQ Password	The password for the above ThreatQ account.
Status	This is the default status for objects that are created by this Integration.

Example Output

```
/opt/tqvenv/<environment_name>/bin/tq-conn-events-builder -ll /var/log/tq_labs/ -c /etc/tq_labs/ -v3
ThreatQ Host: <ThreatQ Host IP or Hostname>
ThreatQ Client ID: <ClientID>
ThreatQ Username: <EMAIL ADDRESS>
ThreatQ 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 entry to open its details page.
4. Enter the following parameters under the **Configuration** tab:

PARAMETER	DESCRIPTION
Private IPs	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.
Filter Out Events	<p>You can filter out certain events based on their attribute names. This value should be in following format:</p> <pre> --- <Syslog source 1>: <tq_object>: <Attribute Name>: <Attribute Value> <Attribute Name>: <Attribute Value> <Syslog source 2>: <tq_object>: <Attribute Name>: <Attribute Value> </pre> <p>In the example above, the <tq_object> is one of the following - attributes, adversaries, asset, attack_pattern, campaign, corporate_email, identity, incident, indicators, malware, uri_source, uri_target, or vulnerability. The <Attribute Name> would either be exactly an attribute name, a TQ object type (as the URI type) - ie. Identity - or an Indicator Type, ie. IP Address; <Attribute Value> would simply be the value. If a TQ object has no type, <Attribute Name>: <Attribute Value> would simply be replaced by the value - see malware below below.</p> <p>If you wish to pass Event Time 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>

PARAMETER	DESCRIPTION
	<p>Example with Sources:</p> <pre> --- McAfee EP0: uri_target: Identity: SYSTEM attributes: Source Process: java.exe malware: RAT McAfee EP0: attributes: Event Time: 2020-10-01 12:57:18 Workstation Name: 123AEP, 436asd, sdf345 </pre>
Events Aggregation Rules by Source	<p>List the aggregation rules by Event source. These are the aggregation rules the integration will use to decide when to create a new event and when to keep adding to an existing event. Do not include a source, if there are no aggregation rules. The format is as follows:</p> <pre> --- <Syslog source 1>: relationships: <Object Type>,<Object Type> <Syslog source 2>: relationships: <Object Type>,<Object Type> </pre> <p>Example:</p> <pre> --- Fidelis Extended: relationships: URI Sources,URI Targets QRadar: relationships: URI Sources,URI Targets </pre>
Update Event Attributes by Source	<p>List the names of the attribute for an Event that need to be updated. Note that only the Event attributes can be updated. Do not include a source, if there is no need to update any attributes from that source. The format is as follows:</p> <pre> --- <Syslog source 1>: attributes: <Attribute Name>,<Attribute Name>,<Attribute Name> Syslog source 2>: </pre>

PARAMETER

DESCRIPTION

attributes: <Attribute Name>,<Attribute Name>,<Attribute Name>,<Attribute Name>

Example:

Fidelis Extended:
attributes: Severity,Status Code,Threat Score,Alert ID

QRadar:
attributes: Country Code,Threat Score,Offense ID

Configuration

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 integration documentation for the format of this field.

Events Aggregation Rules By Source

Fidelis Extended:
relationships: URI Sources,URI Targets
QRadar:
relationships: URI Sources,URI Targets

List the aggregation rules by Event source. Please see the integration documentation for the format of this field.

Update Event Attributes By Source

Fidelis Extended:
attributes: Severity,Status Code,Threat Score,Alert ID

List the names of attributes to be updated by Event source. Please see the integration documentation for the format of this field.

Save

5. Review any additional settings, 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
<code>-h, --help</code>	Shows this help message and exits.
<code>-ll LOGLOCATION, --loglocation LOGLOCATION</code>	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).
<code>-c CONFIG, --config CONFIG</code>	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.)
<code>-v {1,2,3}, --verbosity {1,2,3}</code>	This is the logging verbosity level where 3 means everything. The default setting is 1 (Warning).
<code>-n, --name</code>	This allows you to change the name of the connector.
<code>-d, --no-differential</code>	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 exports after the test. THIS SHOULD NEVER BE USED IN PRODUCTION.

ARGUMENT	DESCRIPTION
<code>-ep, --external-proxy</code>	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.
<code>-p PAYLOAD, --payload PAYLOAD</code>	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 1.2.0**
 - Added the following new configuration parameters:
 - **Events Aggregation Rules by Source** - specify aggregation rules for events.
 - **Update Event Attributes by Source** - specify which event attributes to update.
 - Updated the logic to check if the **Point of Contact** for existing events is assigned every time the integration is executed.
 - Added the Event Occurrence attribute to every event. This attribute will count the number of times an event occurs.
- **Version 1.1.0**
 - Updated delimiters to allow parsing of lists.
 - Added clean up logic for emails.
 - The integration now handles cases when the syslog messages from Fidelis provides comma-delimited lists of MITRE Techniques and Names.
- **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