ThreatQuotient

Qualys Scanner Connector Guide

Version 1.2.4

April 06, 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: `1.2.4`
- Compatible with ThreatQ versions >= `4.34.0`
- Python: `3.6`
- Compatible with Qualys versions: `3.0 - 3.10`
- Third-Party Application Hosting Type:
  - Qualys Public Cloud
  - Qualys Private Cloud

Introduction

The Qualys Scanner connector integrates ThreatQ with a Qualys appliance, either cloud-based or on-prem. The vulnerabilities scanner connector collects information about Qualys scans executed in the past days, collects all CVEs related to those vulnerabilities and ingests them in ThreatQ.

The configuration of the connector in the ThreatQ UI gives the user the ability to define the number of historical days for the first run, and every consecutive run searches only for vulnerabilities in scans executed after the last run of the connector.

Additionally, the user can provide the following parameters to filter down the assets found in Qualys scans; Range or a list of IP addresses to search for, Asset group IDs, and minimum severity of vulnerability scans.

Prerequisites

The following items before attempting to install and use the integration.

Requirements

The following items are required for the integration:

- A Qualys Knowledgebase account in order to convert QIDs to CVEs.
- A dedicated account in Qualys with a Manager role.

Time Zone Settings

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
```

Asset Custom Object Installation

The Asset custom object is required if you opt to use the **Automate Assets** optional feature. ThreatQuotient provides a script that will copy the SVG icon and JSON definition file to the correct directories and automatically install the Asset custom object. These files are bundled with whl file on the ThreatQ Marketplace. You must download this zip file from the ThreatQ Marketplace, even if you are installing the connector via PIP, in order to download the required custom object files.

> When installing the custom objects, be aware that any in-progress feed runs will be cancelled, and the API will be in maintenance mode.

1. Download the integration zip file from the ThreatQ Marketplace and unzip its contents.
2. SSH into your ThreatQ instance.
3. Install the custom objects using the one of the following methods:
4. Navigate to tmp directory:

   ```
   <> cd /tmp/
   ```

5. Create a new directory:

   ```
   <> mkdir qualys_scanner
   ```

6. Upload the **asset.json**, **asset.svg**, and **install.sh** script into this new directory.
7. Navigate to the new directory, **/tmp/qualys_scanner**, if you have not done so yet.

   The directory should resemble the following:

   - tmp
     - **qualys_scanner**
       - **asset.json**
       - **install.sh**
       - **asset.svg**

8. Run the following command:

```
<> sudo bash install.sh
```

> You must be in the directory that houses the install.sh, SVG, and json files when running this command.

The installation script will automatically put the application into maintenance mode, move the files to their required directories, install the custom object, update permissions, bring the application out of maintenance mode, and restart dynamo.

```
Installing Custom Objects - Step 1 of 5 (Entering Maintenance Mode)

Application is now in maintenance mode.
Installing Custom Objects - Step 2 of 5 (Installing the Asset Custom Object)
Installing Custom Objects - Step 3 of 5 (Configuring image for Asset Custom
Object)
Installing Custom Objects - Step 4 of 5 (Updating Permissions in ThreatQ)
Installing Custom Objects - Step 5 of 5 (Exiting Maintenance Mode)

Application is now live.
```

9. Remove the temporary directory, after the custom object has been installed, as the files are no longer needed:

```
<> rm -rf qualys_scanner
```

> You may need to clear your web browser's cache in order to see Asset in the ThreatQ UI.

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	>= 1.8.4	N/A
threatqcc	>= 1.4.1	N/A
urllib3	>=1.26.9	N/A
python-dateutil	>= 2.8.2	N/A
future	>= 0.18.2	N/A

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
```

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_qualys_scanner
   ```

   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 from a separate device with internet access:

   ```
   <> mkdir /tmp/tq_conn_qualys_scanner

      pip download tq_conn_qualys_scanner -d

      /tmp/tq_conn_qualys_scanner/
   ```

   b. Archive the folder with the .whl files:

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

   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_qualys_scanner.tgz
```

e. Activate the virtual environment:

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

f. 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_qualys_scanner-<version>-
   py3-none-any.whl --no-index --find-links /tmp/conn/
```

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

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-qualys-scanner -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.
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 User in the ThreatQ System for integrations.
ThreatQ Password	The password for the above ThreatQ account.

Example Output

```
/opt/tqvenv/<environment_name>/bin/tq-conn-qualys-scanner -v3 -ll /var/log/tq_labs/ -c /etc/tq_labs/
ThreatQ Host: <ThreatQ Host IP or Hostname>
ThreatQ Client ID: <ClientID>
ThreatQ Username: <EMAIL ADDRESS>
ThreatQ Password: <PASSWORD>
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 parameters under the **Configuration** tab:

PARAMETER	DESCRIPTION
IP/Hostname	The Hostname or IP address of Qualys. The following are common Qualys Hostnames that can be used in this field: ◦ **Qualys US Platform 1**: https://qualysapi.qualys.com ◦ **Qualys US Platform 2**: https://qualysapi.qg2.apps.qualys.com ◦ **Qualys US Platform 3**: https://qualysapi.qg3.apps.qualys.com ◦ **Qualys EU Platform 1**: https://qualysapi.qualys.eu ◦ **Qualys EU Platform 2**: https://qualysapi.qg2.apps.qualys.eu ◦ **Qualys India Platform 1**: https://qualysapi.qg1.apps.qualys.in ◦ **Qualys Private Cloud Platform**: https://qualysapi.<customer_base_url>
Port	The Port for communicating with the Qualys API.
Username	Your Username for logging to Qualys.

PARAMETER	DESCRIPTION
Password	The password associated with the account referenced above.
Number of days for initial run	Enter the number of days of history do you want to search for vulnerability scans for the initial run.
Number of Quality IDs to search at a time	Enter the number of Qualys IDs to search at a time. Consult the administrative setting in Qualys to determine this value. The default value is 1000.
Automate Assets	Select this option if you want to enable the automatic ingestion of assets into you TQ instance. ⚠ This option requires that you have the **Asset** custom object installed on your instance.
Provide a range of IPs to collect data for	Optional - One or more IPs or ranges may be specified. Multiple entries should be comma-separated.
Enter asset groups	Optional - Collect scan information only for hosts belonging to asset groups with certain IDs. One or more asset group IDs and/or ranges may be specified. Multiple entries should be comma-separated
Collect only scans which have specific severities	Optional - Show only detection records which have certain severities. One or more levels may be specified. Multiple entries should be comma-separated.

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-qualys-scanner -c /etc/
   tq_labs/ -ll /var/log/tq_labs/ -v3 VERBOSITY_LEVEL
```

Command Line Arguments

This connector supports the following custom command line arguments:

ARGUMENT	DESCRIPTION
`-h`, `--help`	Shows this help message and exits.
`-n NAME, --name Name`	This sets the name for this connector. In some cases, it is useful to have multiple connectors of the same type executing against a single TQ instance. For example, the Syslog Exporter can be run against multiple target and multiple exports, each with their own name and configuration
`-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 exports after the test. THIS SHOULD NEVER BE USED IN PRODUCTION.
`-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.)

ARGUMENT	DESCRIPTION
`-v {1,2,3}, --verbosity {1,2,3}`	This is the logging verbosity level. The default setting is **1** (Warning). The recommended value is **3** (debug).
`-ds, --disable_ssl`	This allows you to disable SSL verification to all requests to the API.
`-ep, -external-proxy`	This allows you to use the proxy that is specified in the ThreatQ UI.

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 * * * /opt/tqvenv/<environment_name>/bin/tq-conn-qualys-
      scanner -c /etc/tq_labs/ -ll /var/log/tq_labs/ -v3
   ```

4. Save and exit CRON.

Change Log

- **Version 1.2.4**
  - Fixed a custom asset error.
- **Version 1.2.3**
  - Improved integration performance by reducing API requests and bulk uploading.
- **Version 1.2.2**
  - Fixed two bugs related to handling API responses.
- **Version 1.2.1**
  - Added the QIDs as Vulnerability objects in TQ.
  - Related the QIDs to CVEs that are ingested.
  - Added IPs as Asset objects to TQ.
  - Related Assets to QIDs and CVEs.
  - Added a checkbox in the TQ UI config allowing users to enable/disable the automated adding of assets.
  - Unrelated Assets from QIDs and corresponding CVEs if QID was patched/fixed.
- **Version 1.2.0**
  - Changed authentication to basic auth with username and password.
  - Modified the the QID to CVE mapping to do it in batches of 1000 QIDs.
- **Version 1.1.1**
  - Added unicode string handling for the response from Qualys.
- **Version 1.1.0**
  - Added Python 3 support.
- **Version 1.0.0**
  - Initial Release