

ThreatQuotient



Qualys Scanner Connector Guide

Version 1.2.2

May 18, 2021

ThreatQuotient

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

Support

Email: support@threatq.com

Web: support.threatq.com

Phone: 703.574.9893

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 © 2021 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.

Contents

Versioning.....	4
Introduction.....	5
Prerequisites	5
Installation	6
Configuration.....	9
Usage.....	11
Command Line Arguments	11
CRON	13
Change Log.....	14

Versioning

- Current integration version: 1.2.2
- Supported on ThreatQ versions \geq 4.34.0

There are two versions of this integration:

- Python 2 version
- Python 3 version

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 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.

Installation

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. 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:

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



```
pip install /tmp/conn/ tq_conn_qualys_scanner-1.1.0-py2-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 `/usr/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:

```
<> 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.
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.

Example Output

```
tq-conn-qualys-scanner -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 parameters under the **Configuration** tab:

PARAMETER	DESCRIPTION
IP/Hostname	The Hostname or IP address of Qualys.
Port	The Port for communicating with the Qualys API.
Username	Your Username for logging to Qualys.
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.

PARAMETER	DESCRIPTION
 Make sure you have an Asset custom object in your instance before checking this.	
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:

```
<> 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
<code>-h, --help</code>	Shows this help message and exits.
<code>-n NAME, --name Name</code>	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
<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.
<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.)

ARGUMENT	DESCRIPTION
<code>-v {1,2,3}, --verbosity {1,2,3}</code>	This is the logging verbosity level. The default setting is 1 (Warning). The recommended value is 3 (debug).
<code>-ds, --disable_ssl</code>	This allows you to disable SSL verification to all requests to the API.
<code>-ep, -external-proxy</code>	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 * * * tq-conn-qualys-scanner -c /etc/tq_labs/ -ll /var/  
log/tq_labs/ -v3
```

4. Save and exit CRON.

Change Log

- **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