ThreatQuotient

ThreatQ Slackbot Connector Guide

Version 1.0.0

June 13, 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

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.

Integration Details

ThreatQuotient provides the following details for this integration:

Current Integration Version	1.0.0
Compatible with ThreatQ Versions	>= 5.6.0
Python Version	3.6
Third-Party Application Hosting Type	Cloud
Support Tier	ThreatQ Supported

Introduction

The ThreatQ Slackbot Connector enables analysts to interact with ThreatQ via their Slack workspace.

The connector has the ability to perform automatic as well as manual actions using emoji reactions and slash commands.

The connector ingests indicators into the ThreatQ platform.

Prerequisites

Review the following requirements before attempting to install the connector.

Third-Party Credentials

The following credentials are required by the connector:

- The Bot User OAuth Token used to authenticate with Slack.
- The App-Level Token used to connect to ThreatQ Slack Application.

Creating the ThreatQ Slackbot

You must first create your slackbot within your Slack workspace before you can install the connector.

1. Navigate to your Slack Apps: https://api.slack.com/apps.
2. Click the **Create New App** button
3. Select the **From an App Manifest** option when prompted to select how to configure your app's scopes and settings.
4. Select the workspace you want to develop your app in and click on **Next**.

   > This is typically the one you want to install it in as well.

5. When prompted to enter your manifest, select the **YAML** tab and then paste the YAML below:

```
_metadata:
  major_version: 1
  minor_version: 1
display_information:
  name: ThreatQ
  description: The ThreatQ App for Slack enables users to interact with ThreatQ via Slack.
  long_description:
    The ThreatQ App for Slack enables Slack users to interact with ThreatQ
    in various ways to facilitate their intelligence workflows. This includes performing
    automated lookups, submitting indicators, querying ThreatQ, and more.
  background_color: "#0055AA"
settings:
  socket_mode_enabled: true
  interactivity:
    is_enabled: false
  event_subscriptions:
    bot_events:
      - message.channels
      - message.groups
      - message.im
      - message.app_home
      - reaction_added
features:
  app_home:
    home_tab_enabled: false
    messages_tab_enabled: true
    messages_tab_read_only_enabled: false
  bot_user:
    display_name: ThreatQ
    always_online: true
  slash_commands:
    - command: /tq-help
      description: Displays the available commands for ThreatQ.
      usage_hint: /tq-help
    - command: /tq-ioc
      description: Performs a lookup for an IOC in ThreatQ.
      usage_hint: /tq-ioc <indicator>
    - command: /tq-create-ioc
      description: Creates an IOC in ThreatQ.
      usage_hint: /tq-create-ioc <type> <indicator>
    - command: /tq-reload-user-config
      description: This will reload the user configurations for this integration.
oauth_config:
  scopes:
    bot:
      - commands
      - chat:write
      - chat:write.public
      - im:read
      - im:write
      - im:history
      - groups:read
      - groups:history
      - channels:read
      - channels:history
      - metadata.message:read
      - reactions:read
      - reactions:write
```

6. Review the app summary and then click on **Create**.

   > You will be redirected to the App's settings page

7. Scroll down to your App's Display Information, apply the ThreatQ logo, and then click on **Save Changes**.

   > You can find the ThreatQ logo (rhino head) here: https://www.threatq.com/branding/

8. Click on the Install Your App option under the Building Apps for Slack section.
9. Select the workspace to install the app into.

10. Click on the **Generate Token and Scopes** button under the App-Level Tokens section.
11. Name your token, click the **Add Scope** button, select the **connections:write** scope, and then click on **Generate**.
12. Copy and Save the **Token** to a secure location.
13. Click on the **Install App** option under the Settings section.
14. Copy and Save the **Bot user OAuth Token** to a secure location.

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
threatqcc	>=1.4.2	N/A
python-dateutil	>=2.8.2	N/A
slack-bolt	**1.15.0**	**Pinned**

Installation

⚠ You must first create the slackbot before you can install the connector.

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 python-dateutil slack-bolt
pip install setuptools==59.6.0
```

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_slackbot-<version>-py3-none-any.whl
   ```

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

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-slackbot -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.

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

Example Output

```
/opt/tqvenv/<environment_name>/bin/tq-conn-slackbot -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>
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
Bot User OAuth Token	Enter the Bot User OAuth Token (xoxb-_) to authenticate with Slack.
App-Level Token	Enter the App-Level Token (xapp-_) to authenticate with Slack.
ThreatQ Hostname / IP Address	Enter the hostname or IP address of your ThreatQ instance.
Create IOCs Using the ➚ Reaction	Enabling this option will allow users to manually parse & upload IOCs by reacting to a message with the ➚ emoji.
Lookup IOCs Using the 🔍 Reaction	Enabling this option will allow users to manually lookup IOCs by reacting to a message with the 🔍 emoji.
Automatically Perform Lookups for the Following Types	Select the types that you would like to automatically perform lookups for. Lookup results will be returned in a thread for the message that contained the value.

PARAMETER	DESCRIPTION
Automatically Create IOCs for Detected Types	Select the IOC types that you would like to automatically create when detected in Slack. Creation results will be returned in a thread for the message that contained the IOC.

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-slackbot -v3 -ll /var/
log/tq_labs/ -c /etc/tq_labs/
```

Slash Commands

The following slash commands are available for the application:

COMMAND	DESCRIPTION	EXAMPLE
/tq-help	Displays the available commands for ThreatQ	/tq-help
/tq-ioc <indicator>	Performs a lookup for an IOC in ThreatQ	/tq-ioc be2b25c82d2e0d3c5736c4382bf1de3d895cb2e1
/tq-create-ioc <type> <indicator>	Creates an IOC in ThreatQ	/tq-create-ioc "IP Address" 8.8.8.8
/tq-reload-user-config	This will reload the user configurations for this integration	/tq-reload-user-config

The following capabilities can be enabled from the user configuration:

- Create IOCs using the ⤴ emoji: After reacting to a message with the ⤴ emoji, it will be parsed and all the IOCs found are uploaded to ThreatQ.

- Lookup IOCs using the 🔍 emoji: After reacting to a message with the 🔍 emoji, it will parsed and all the IOCs found will be searched in ThreatQ.
- Automatically perform lookups: Automatically search through all the messages sent after this option is enabled and bring information from ThreatQ about all the IOCs that are found. This depends on the selected types.
- Automatically create IOCs: The messages sent after this option is enabled will be parsed and any contained IOCs will be uploaded to ThreatQ. This depends on the selected types.

Command Line Arguments

This connector supports the following custom command line arguments:

ARGUMENT	DESCRIPTION
`-h`, `--help`	Review all additional options and their descriptions.
`-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.
`-n, --name`	Optional - Name of the connector (Option used in order to allow users to configure multiple connector instances on the same TQ box).

Setup as a Service

This integration is an always-on integration, meaning it does not run on a schedule. Instead you must create a service for the connector so that it is always running.

To setup the integration as a service, follow the steps below:

1. Log into your ThreatQ host via a CLI terminal session.
2. Create a file for the new service

```
sudo vi /etc/systemd/system/tq_conn_slackbot.service
```

3. Copy and paste the following into the file:

```
[Unit]
Description=ThreatQ Slackbot Connector
# Require the network to be up before starting this service
After=network.target
[Service]
Type=simple
User=root
WorkingDirectory=/root
ExecStart=/opt/tqvenv/slackbot/bin/tq-conn-slackbot -v 3 -c /etc/tq_labs/ -ll /var/log/tq_labs/
Restart=always
TimeoutSec=10
[Install]
WantedBy=multi-user.target
```

4. Save and exit the file (`:wq`).
5. Give the service file the correct permissions:

```
sudo chmod 664 /etc/systemd/system/tq_conn_slackbot.service
```

6. Reload the systemd daemon:

```
sudo systemctl daemon-reload
```

7. Start the service:

```
sudo systemctl start tq_conn_slackbot.service
```

8. Enable the service to start on boot

```
sudo systemctl enable tq_conn_slackbot.service
```

Startup Troubleshooting

If you are having issues with the service, you can check the status of the service by running the following command:

```
sudo systemctl status tq_conn_slackbot.service
```

You can also check the logs by running the following command:

```
sudo journalctl -fu tq_conn_slackbot.service
```

Updating Slackbot Configuration

Use the follow steps if you have to update the slackbot:

1. Log into the Slack API portal and navigate to your apps: https://api.slack.com/apps/
2. Select the **ThreatQ App**.
3. Navigate to the App Manifest section and copy/paste the latest YAML configuration from this guide - see step 5 in the Creating the ThreatQ Slackbot section.
4. Click on **Save Changes**.
5. Navigate to the Install App section, click on the **Reinstall App to Workspace** option and follow all prompts.

Known Issues / Limitations

- The Slackbot will not return any result for incorrect commands.

  For example, the following command will not produce a reply message: `/tq-create-ioc IP Address 8.8.8.8` . This is because the command is incorrect due to the IOC types that contain spaces should be placed in double quotes or the spaces can be replaced with '_'.

- The Slackbot does not work with Unix file paths (e.g /tmp/folder/file.txt) because it will try to interpret them as Slack commands. Only Windows file paths are allowed. (e.g C:\myfolder\file.txt, \server\share\file.txt).
- Email addresses, URLs and FQDNs from certain whitelisted domains (e.g gmail.com, facebook.com, github.com) are not uploaded to ThreatQ.
- Automatic IOC lookups and creations and searching IOCs using 🔍 emoji work only with one IOC per message. For example, sending this message: `d41d8cd98f00b204e9800998ecf8427a 8.8.8.8` only the MD5 hash will be automatically created/looked up.
- Creating IOCs using ↗ emoji works with multiple IOCs per message. For example, sending this message `d41d8cd98f00b204e9800998ecf8427a 8.8.8.8` and reacting with ↗ will upload both, the MD5 and the IP Address to ThreatQ.

Change Log

- **Version 1.0.0**
    - Initial release