ThreatQuotient

ThreatQ Connector for Microsoft Exchange Online

Version 1.0.0

March 15, 2024

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	>= 4.40.0
Python Version	3.6
Support Tier	ThreatQ Supported

Introduction

The ThreatQ Connector for Microsoft Exchange Online is a unidirectional integration that integrates directly with the Microsoft Exchange mail server. The integration enables the ingestion of emails and attachments into ThreatQ and supports parsing emails and attachments as well as forwarded emails (spearphishing).

The connector uses the following endpoints:

- **Auth -** used to authenticate.
- **Messages -** used to fetch all the messages and to mark them as read.
- **Attachments -** used to fetch all the messages attachments.

The integration ingests the following system objects:

- Events
- Indicators
- Attachments

Prerequisites

Review the following requirements before attempting to install the connector.

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`

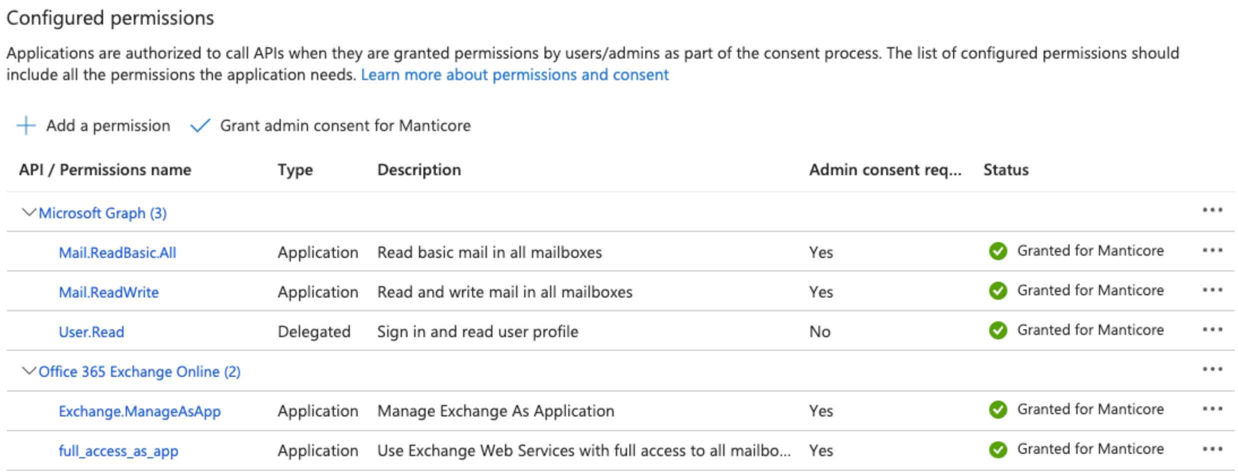
Azure Portal Active Directory Registration

The integration requires a registration in the Azure Portal Active Directory. Use the following steps to register the app:

1. Go to portal.azure.com and log in as an administrator.
2. Click on **Azure Active Directory** and then on A**pp registrations**.
3. Select **New registration**.
4. Enter the **Name** and select **Accounts** in this organizational directory only and then click **Register**.
5. Click on **API permissions**, select **Add a permission** and on the next page click on the tab **APIs my organization uses**.
6. Enter the following permissions one-by-one

API CATEGORY	API/PERMISSION NAME	PERMISSION TYPE
Microsoft.Graph	User.Read	Delegated
Microsoft.Graph	Mail.ReadWrite	Application
Microsoft.Graph	Mail.ReadBasic.All	Application
Office 365 Exchange Online	Exchange.ManageAsApp	Application
Office 365 Exchange Online	full_access_as_app	Application

⚠ All Application type permissions need to be granted consent to the organization by clicking on the checkmark **Grant admin consent for <Org Name>** right above the table.



Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. Learn more about permissions and consent

API / Permissions name	Type	Description	Admin consent req...	Status	
Microsoft Graph (3)					•••
Mail.ReadBasic.All	Application	Read basic mail in all mailboxes	Yes	✓ Granted for Manticore	•••
Mail.ReadWrite	Application	Read and write mail in all mailboxes	Yes	✓ Granted for Manticore	•••
User.Read	Delegated	Sign in and read user profile	No	✓ Granted for Manticore	•••
Office 365 Exchange Online (2)					•••
Exchange.ManageAsApp	Application	Manage Exchange As Application	Yes	✓ Granted for Manticore	•••
full_access_as_app	Application	Use Exchange Web Services with full access to all mailbo...	Yes	✓ Granted for Manticore	•••

> Once completed your **API permissions** should look similar to example image above. The Status column should show green checkboxes instead of orange warning triangles. If you see the warning triangles, it means that the permissions have not been granted consent by your organization's Azure administrator.

7. Obtain credentials (Client ID, Client Secret and Tenant ID) by clicking on **Certificates and secrets** and then on **New client secret**.
8. Enter a **Description** for the credentials set and then click on **Add**. You will be taken to a new page and make sure to save the **Client Secret** which is in the **Value** column.
9. Click on **Overview** on the left menu and copy the **Client ID** and **Tenant ID**.
10. Save the **Client Secret**, **Client ID** and **Tenant ID** as they will be needed for the final configuration of the Microsoft Exchange Online connector.

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	>=1.8.6	N/A
threatqcc	>= 1.4.2	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 threatqsdk threatqcc
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_ms_exchange_online-<version>-py3-none-any.whl
```

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

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-ms-exchange-online -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.
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-ms-exchange-online -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
Email Address	The email address for the Office 365 Online Exchange email box you want to ingest email from.
Client ID	The client ID for the application. This is available from Microsoft Azure.
Client Secret	The client secret for the application. This is available from Microsoft Azure.
Tenant ID	The tenant ID of the application. This is available from Microsoft Azure.
Mailbox Folders	Enter a comma-separated list of folders to pull emails from. The default value setting is **Inbox**.
Mailbox Type	Select the mailbox type. Options include: ◦ **Intelligence** - This will import and parse all emails and attachments for indicators. ◦ **Spearphish** - This will only parse emails with spearphish attachments (.eml, .emlx, .pst, .ost).
Trim number of characters	The number of characters to trim from the email body. ⚠ The Trim Body parameter should be used with caution. Useful indicator data could be trimmed out resulting in ingested email files having an invalid format.

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-ms-exchange-online -v3 -ll /
var/log/tq_labs/ -c /etc/tq_labs/
```

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.
`-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.
`--external-proxy, -ep`	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.

ARGUMENT	DESCRIPTION
`-ds, --disable-ssl`	Disable SSL verification.

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-ms-exchange-online -c /etc/tq_labs/ -ll /var/log/tq_labs/ -v3
   ```
4. Save and exit CRON.

Known Issues / Limitations

- The Trim Body parameter should be used with caution. Useful indicator data could be trimmed out resulting in ingested email files having an invalid format.

Change Log

- **Version 1.0.0**
    - Initial release