

ThreatQuotient

A Securonix Company



Maltego TRX for ThreatQ

Version 2.1.0

November 12, 2025

ThreatQuotient

20130 Lakeview Center Plaza Suite 400
Ashburn, VA 20147

 ThreatQ Supported

Support

Email: tq-support@securonix.com

Web: <https://ts.securonix.com>

Phone: 703.574.9893

Contents

Warning and Disclaimer	3
Support	4
Integration Details.....	5
Introduction	6
Prerequisites	7
Generating ThreatQ OAuth Credentials	8
ThreatQ v6 Steps	8
ThreatQ v5 Steps	9
Installation.....	10
Installing into a TDS Instance	10
Running the ThreatQ TRX App.....	10
Generating the Artifacts	12
Setting Up the TDS Instance	12
Configuring the Transforms in your Maltego Client.....	13
Installing Locally into Maltego Using Docker	14
Installing Locally into Maltego In a Python Virtual Environment	17
Docker Shared Directories	20
Transforms	21
Upgrading the App	22
Migrating from Previous Versions	23
Development.....	24
Testing with a TDS Instance.....	24
Packaging the TRX App for Distribution.....	25
Known Issues / Limitations	26
Change Log	27

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 © 2025 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: tq-support@securonix.com

Support Web: <https://ts.securonix.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 2.1.0

**Compatible with ThreatQ
Versions** >= 5.6.0

Python Version >= 3.10

Third-Party Application Maltego Desktop Application

Support Tier ThreatQ Supported

Introduction

The Maltego TRX Integration for ThreatQ is a comprehensive transform library that enables seamless bi-directional communication between ThreatQ and Maltego. It empowers analysts to perform contextual lookups for indicators of compromise (IOCs) and other object types—such as relationships, tags, and attributes—providing deeper visibility into their ThreatQ threat intelligence repository.

With over 450 transforms, the integration allows interaction with a wide range of IOC types and custom entities representing objects within ThreatQ. Additionally, it offers compatibility with select entities from the official STIX2 plugin, further enhancing analytical capabilities within Maltego.

Prerequisites

The following is required by the integration:

- ThreatQ Login Credentials. Options authenticated methods are accepted:



You cannot use user-based authentication if the user account was created through SSO or if the user account has MFA/2-step verification enabled. You must use the OAuth Client Credentials authentication method in those instances.

- **User-Based Authentication** - Your ThreatQ Username, Password, and Client ID that be located under your ThreatQ user profile.
- **OAuth Client Credentials** - [OAuth Client ID and Secret generated via ThreatQ CLI commands](#).
- Maltego Desktop Client - you can download Maltego for free at: <https://www.maltego.com/downloads/>.
- Installing into a TDS Instance specific:
 - Transforms Server with Docker.



If you do not have Docker installed, you can find the installation instructions at: <https://docs.docker.com/get-docker/>

- TDS instance.
- The ThreatQ integration files (kept in their respective directories):



These integration files are bundled together and can be downloaded from the ThreatQ Marketplace under the [Maltego TRX for ThreatQ](#) entry.

- `artifacts/mtz/securonix-entities.mtz`
- `artifacts/csv/securonix-transforms.csv`
- `artifacts/csv/securonix-settings.csv`
- `dependencies/threatqsdk-*-py3-none-any.whl`
- `dependencies/maltego_threatq_trx-*-py3-none-any.whl`
- `docker-compose.yml`
- `app.env`
- Installing Locally into Maltego Using Docker specific:
 - Docker - you can find installation instructions at: <https://docs.docker.com/get-docker/>
 - The following integration files (kept in their respective directories):
 - `dependencies/threatqsdk-*-py3-none-any.whl`
 - `dependencies/maltego_threatq_trx-*-py3-none-any.whl`
 - `docker-compose.yml`
 - `app.env`
- Installing Locally into Maltego Into a Python Virtual Environment specific:
 - A Python v3.10+ Virtual Environment.
 - The following integration files (kept in their respective directories):

- `dependencies/threatqsdk-*-py3-none-any.whl`
- `dependencies/maltego_threatq_trx-*-py3-none-any.whl`

Generating ThreatQ OAuth Credentials

You will need to generate oauth2 client credentials in order to successfully have the app authenticate with ThreatQ. This is performed using the command line of the ThreatQ appliance.

ThreatQ v6 Steps

1. SSH to your ThreatQ installation.
2. Create a new client id and client secret password using the following command:

```
kubectl exec --namespace threatq --stdin --tty deployment/api-  
schedule-run -- ./artisan threatq:oauth2-client --name="maltego"
```

3. Copy the `client_id` and `client_secret` from the output for use in the configuration of the ThreatQ Maltego application.

Example Output:

```
session_timeout_minutes: 1440  
name: maltego  
type: private  
client_id: ntdjzwe3mduyyjqxyjdiyza5mzyxmtkx  
client_secret:  
YThl0TBLZjM0YTYxNWM1YjVkdDdmMTdjNGY5MzZkYTg4M2RmYmRiZGJmNjk1OTRm  
updated_at: 2020-01-14 14:03:27  
created_at: 2020-01-14 14:03:27
```


ThreatQ v5 Steps

1. SSH to your ThreatQ installation.
2. Navigate to the api directory using the following command:

```
cd /var/www/api
```

3. Create a new client id and client secret password using the following command:

```
php artisan threatq:oauth2-client --name="maltego"
```

4. Copy the client_id and client_secret from the output for use in the configuration of the ThreatQ Maltego application.

Example Output:

```
session_timeout_minutes: 1440
name: maltego
type: private
client_id: ntdjzwe3mduyyjqxyjdiyza5mzyxmtkx
client_secret:
YThlOTBlZjM0YTYxNWM1YjVkODdmMTdjNGY5MzZkYTg4M2RmYmRiZGJmNjk1OTRm
updated_at: 2020-01-14 14:03:27
created_at: 2020-01-14 14:03:27
```

Installation

Use the following steps to setup and install the integration.

1. Determine your ThreatQ authentication type (local vs OAuth) and collect the required credentials based on your selection. See the [Prerequisites](#) section for more details.
2. Download the integration files from the ThreatQ Marketplace at: <https://marketplace.threatq.com/details/maltego-trx-for-threatq>.
3. Extract the integrations files to a directory on your server/workstation. The folder structure should resemble the following:

```
threatq-maltego-integration/  
├── artifacts  
│   ├── mtz  
│   │   ├── securonix-transforms.mtz  
│   │   └── securonix-entities.mtz  
│   └── csv  
│       ├── securonix-settings.csv  
│       └── securonix-transforms.csv  
├── dependencies  
│   ├── threatqsdk-*-py3-none-any.whl  
│   └── maltego_trx-*-py3-none-any.whl  
├── app.env  
└── docker-compose.yml
```



These files will be required for the installation process on your TDS instance or local Maltego client.

Installing into a TDS Instance

The following steps will instruct you on how to install the ThreatQ transforms into your TDS instance. Review and confirm that you completed all [prerequisites](#) before proceeding.

Running the ThreatQ TRX App

The ThreatQ TRX app must be running in order to use the transforms. You can run the server using docker (recommended) or directly on your machine. The following steps will guide you through running the server using docker:

1. Navigate into the extracted integration files directory in your terminal.
2. Optional - If you want to utilize a single set of credentials that all clients can use, while also not exposing those credentials to clients, you can modify the `app.env` file. You can set the environment variables for user-based authentication or OAuth client credentials. The following variables are available:

VARIABLE	DESCRIPTION
TQ_HOST	The hostname or IP address of your ThreatQ instance. Example: <code>example.threatq.online</code> .
TQ_VERIFY_SSL	Set to <code>true</code> to verify SSL certificates, or <code>false</code> to disable SSL verification. <div>  This is not recommended for production. </div>
TQ_USERNAME	The username for ThreatQ login if using the user-based authentication method.
TQ_PASSWORD	The password for ThreatQ user above if using the user-based authentication method.
TQ_CID	The Client ID for the ThreatQ user if using the user-based authentication method.
TQ_OAUTH_CLIENT_ID	Your OAuth Client ID if using the OAuth authentication method.
TQ_OAUTH_CLIENT_SECRET	Your OAuth Client Secret if using the OAuth authentication method.
<div>  If you do not set these variables, clients will need to provide them when running the transforms. </div>	

3. Build and run the ThreatQ TRX app using Docker Compose:

```
docker compose up -d
```

The server should now be running on port 8080 by default. You can access the server by navigating to `http://<your-server-ip>:8080` in your web browser. If you would like to change the port, you can modify the `docker-compose.yml` file to map a different port on your host machine to port 8080 in the container.

4. Proceed to Generating the Artifacts.

Generating the Artifacts

Once the ThreatQ TRX app is running, you will need to generate the artifacts needed for Maltego. This includes the `.mtz` files and the `.csv` files for the transforms. These artifacts will be used to import the transforms into your TDS instance.



The integration package comes with pre-generated artifacts, but you'll want to regenerate them to ensure they are up-to-date and include your particular transforms host URL.

Run the following command to generate the artifacts:



Replace `<your-transforms-host>` with the hostname or IP address of your Transforms Server. This is what will be used in the `securonix-transforms.csv` file so clients can connect to the transforms server.

macOS/Linux:

```
docker exec -it maltego-threatq-trx maltego-threatq-trx-generate-artifacts
--cmd "$(which docker)" --transforms-host "<your-transforms-host>"
```

Windows:

```
docker exec -it maltego-threatq-trx maltego-threatq-trx-generate-artifacts
--cmd "$(Get-Command docker | Select-Object -ExpandProperty Source)" --cwd
"$(Get-Location)" --transforms-host "<your-transforms-host>"
```



You may see a handful of WARNING logs. They can be ignored. The required files will be generated in the `artifacts/mtz` and `artifacts/csv` directories.

Proceed to [Setting up the TDS Instance](#) section.

Setting Up the TDS Instance

You'll need the integration files that were downloaded from the ThreatQ Marketplace.

1. Log into your TDS instance.
2. Navigate to the Paired Configurations tab and click on the **Add New Paired Configuration** button.
3. Set the **Name** field to Securonix.
4. Use the **Choose File** button to select the `securonix-entities.mtz` package.
5. Click on the **Add Paired Configuration** button to save.
6. Navigate to the Transform Settings tab and click on the **Import Transform Settings** button (up arrow) located at the top right corner.
7. Use the **Choose File** button to select the `securonix-settings.csv` file.
8. Click the **Import Transform Settings** button to save.
9. Navigate to the Seeds tab and click the **Add Seed** button.

10. Set the **Seed Name** field to `Securonix`
11. Set the **Seed URL** field to whatever identifier you want (i.e. `securonix`).
12. Select the `Securonix` paired configuration from the dropdown.
13. Click on the **Add Seed** button to save.
14. Navigate to the **Transforms** tab and click the import button (up arrow) in the top right corner.
15. Use the **Choose File** button to select the `securonix-transforms.csv` file.



If you are upgrading from a previous version, you will need to check the **Override existing rows with the same 'Name' column** checkbox. If you do not, it may create duplicate transforms.

16. Click on the **Import Transform** button to save.



This request may timeout, but the transforms will still be imported. You can verify this by checking the **Transforms** tab.


If you would like to set the default settings for the transforms, you can do so by navigating to the **Transform Settings** tab and clicking the **Edit** button next to each of the transform settings you want to set default values for. If you do not set the default values, users will need to set the values manually when running the transforms. We recommend setting a default value for the 'Securonix Host' setting at a minimum. Keep in mind, the default values you set for settings will be visible to all clients.

17. Proceed to [Configuring the Transforms in your Maltego Client](#).

Configuring the Transforms in your Maltego Client

You will need to configure your local Maltego client to use the transforms once your organization has installed the transforms into your TDS instance.

1. Open Maltego and navigate to the **Transforms** tab.
2. Select the **Maltego Data Hub** option in the tool bar.
3. Scroll to the bottom of the list to find the **Internal Hub Items** section.
4. Click the **+** button to add a new hub item.
5. Fill out the form with the following information:

FIELD	DETAILS
ID	securonix
Name	Securonix
Seed URL	<p><Your TDS instance URL>/runner/showseed/<seed-id></p> <div>  <p>Replace <seed-id> with the identifier that was configured as the Seed URL in the TDS instance (e.g. threatq).</p> <p>Alternatively, copy the URL from the Seeds tab in your TDS instance, replacing the hostname with your TDS instance URL.</p> </div>
Description	Enter Securonix transforms for Maltego.
Icon URL	Download the icon from the following: https://cdn.prod.website-files.com/67672dff14ccdaa30a20f455/67fe7f196994c29ae159c875_ThreatQuotient-2025-rhino-head.png .

6. Click on **OK** to save the hub item.




Once the hub item is created, the transforms and entities should sync to your Maltego client automatically. If they do not, you can manually refresh the transforms by clicking the Refresh button when you mouse-over the ThreatQ hub item you created.

Installing Locally into Maltego Using Docker

The following steps will instruct you on how to install and configure the integration with ThreatQ, using Docker. Please make sure that all the [prerequisites](#) listed below are completed *prior* to starting these steps.

1. Navigate into the extracted integration files directory in your terminal.
2. Locate the `app.env` file and open it in a text editor.
3. Complete the following environment variables based on the authentication method (user-based vs OAuth):

VARIABLE	DESCRIPTION
TQ_HOST	The hostname or IP address of your ThreatQ instance. Example: <code>example.threatq.online</code> .
TQ_VERIFY_SSL	Set to <code>true</code> to verify SSL certificates, or <code>false</code> to disable SSL verification. <div>  This is not recommended for production. </div>
TQ_USERNAME	The username for ThreatQ login if using the user-based authentication method.
TQ_PASSWORD	The password for ThreatQ user above if using the user-based authentication method.
TQ_CID	The Client ID for the ThreatQ user if using the user-based authentication method.
TQ_OAUTH_CLIENT_ID	Your OAuth Client ID if using the OAuth authentication method.
TQ_OAUTH_CLIENT_SECRET	Your OAuth Client Secret if using the OAuth authentication method.

- Save your changes.
- Build and run the ThreatQ TRX app using Docker Compose:

```
docker compose up -d
```

- Run the following command to generate the artifacts needed for Maltego:

macOS/Linux:

```
docker exec -it maltego-threatq-trx maltego-threatq-trx-generate-artifacts --cmd "$(which docker)"
```

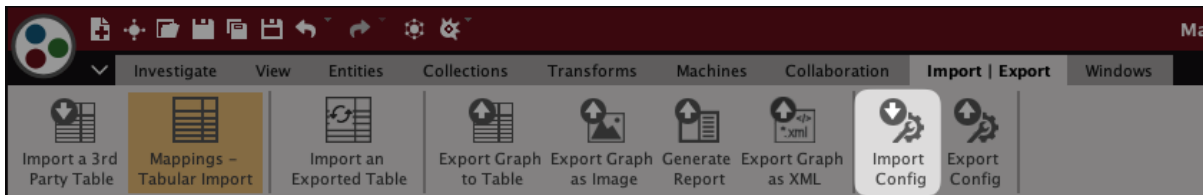
Windows:

```
docker exec -it maltego-threatq-trx maltego-threatq-trx-generate-artifacts --cmd "$(Get-Command docker | Select-Object -ExpandProperty Source)" --cwd "$(Get-Location)"
```



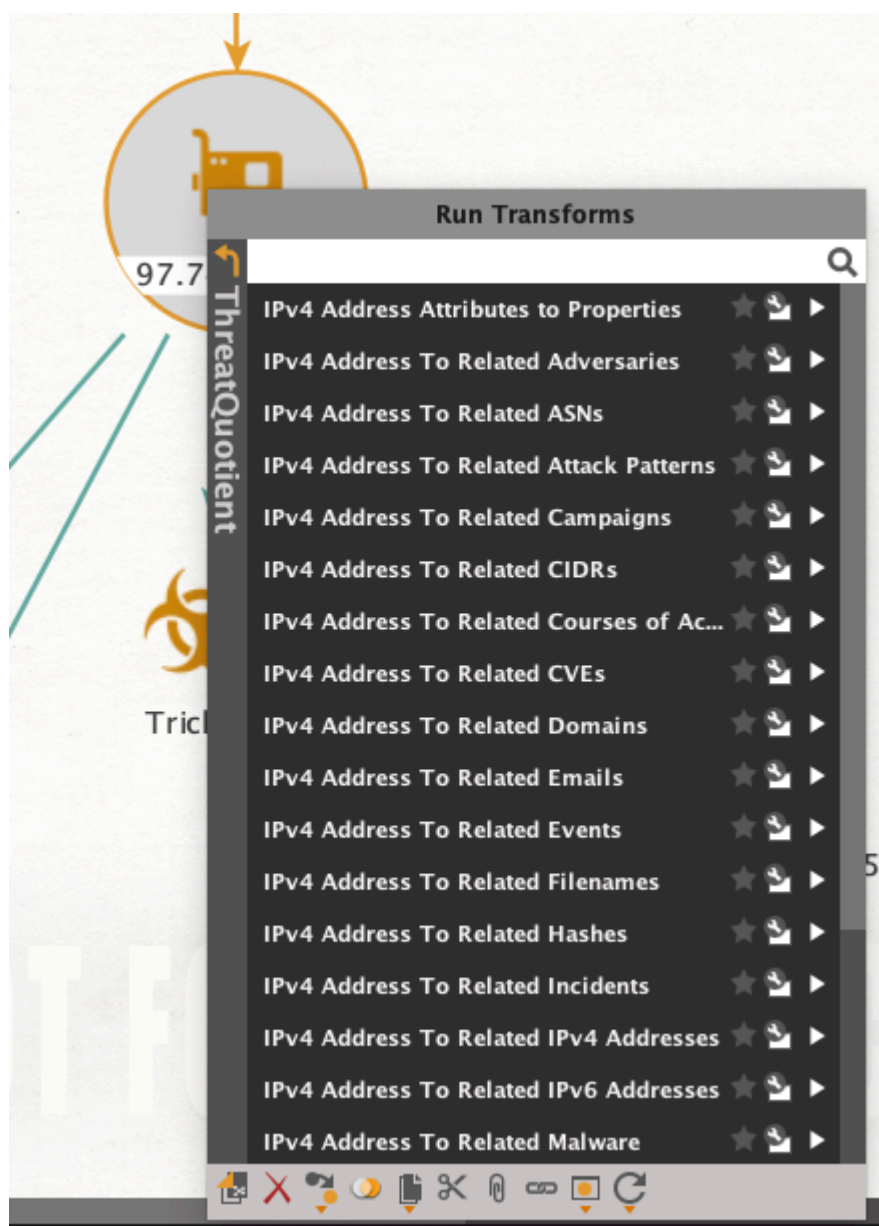
You may see a handful of WARNING logs. They can be ignored. The required files will be generated in the `artifacts/mtz` directory.

7. Open Maltego and navigate to the **Import | Export** tab, and click on the **Import Config** button.



8. Use the file browser to locate the `.mtz` files in `./artifacts/mtz`.
9. Select the `securonix-entities.mtz` package and click on **Next**.
10. Use the checkboxes to select what you want to import and then click on **Next**.
11. Complete the import.
12. Repeat steps 9-11 with the `securonix-transforms.mtz` package.

You should now be able to run the ThreatQ transforms on your Maltego entities



Installing Locally into Maltego In a Python Virtual Environment

The following steps will instruct you on how to install and configure the integration with ThreatQ using a Python virtual environment. Please make sure that all the [prerequisites](#) listed below are completed *prior* to starting these steps.

1. Navigate into the extracted integration files directory in your terminal.
2. Create a configuration file named `threatquotient.cfg` in the current directory.
3. Paste the following content into the `threatquotient.cfg` file and fill out the fields as needed:

```
[threatquotient]
securonix_host=
securonix_verify_ssl=true
```

```
# OAuth Client Credentials
securonix_oauth_client_id=
securonix_oauth_client_secret=
# User Authentication
securonix_username=
securonix_password=
securonix_clientid=
```



Enter either OAuth Client Credentials or User Authentication credentials based on your selected authentication method.

4. Create a Python v3.10 or newer virtual environment in the current directory.
5. Install Python 3.10 or newer if you haven't already.

```
python3 -m venv venv
```

6. Activate the virtual environment:

macOS/Linux:

```
source venv/bin/activate
```

Windows:

```
.\venv\Scripts\Activate.ps1
```

7. Install the required dependencies from the dependencies directory:

```
pip install dependencies/threatqsdk-*-py3-none-any.whl
pip install dependencies/maltego_threatq_trx-*-py3-none-any.whl
```

8. Run the following command to generate the artifacts needed for Maltego:

macOS/Linux:

```
maltego-threatq-trx-generate-artifacts --cmd "$(which maltego-threatq-trx)" --cwd "$(pwd)" --params ""
```

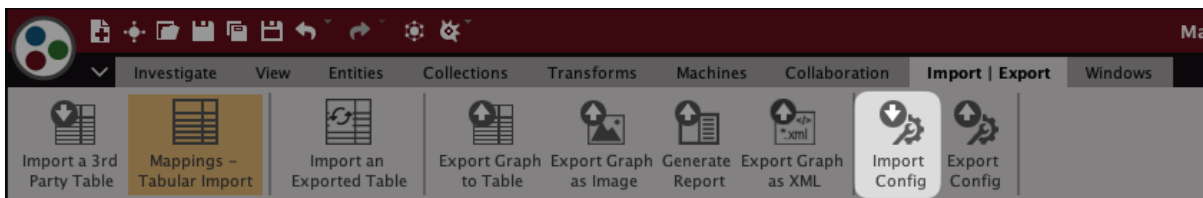
Windows:

```
maltego-threatq-trx-generate-artifacts --cmd "$(Get-Command maltego-threatq-trx | Select-Object -ExpandProperty Source)" --cwd "$(Get-Location)" --params ""
```



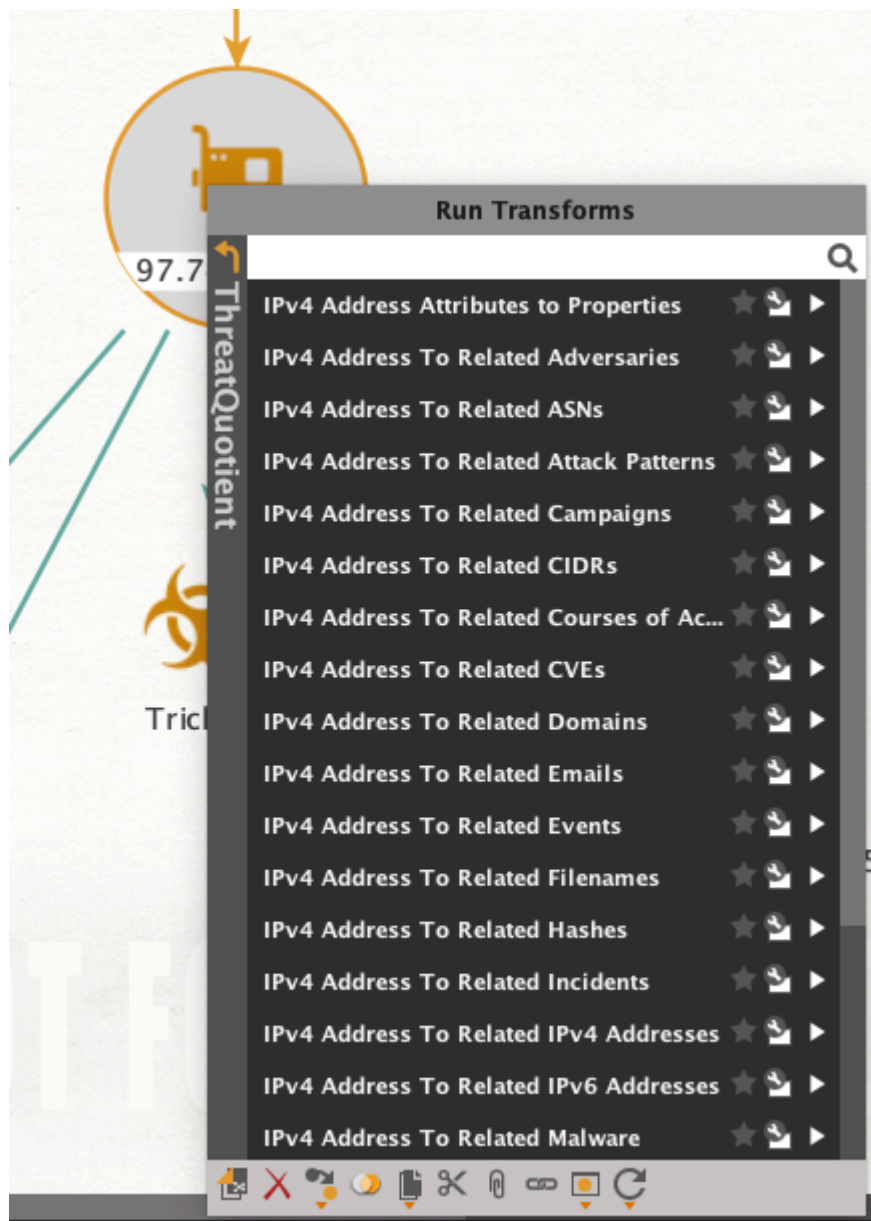
You may see a handful of WARNING logs. They can be ignored. The required files will be generated in the artifacts/mtz directory.

- Open Maltego and navigate to the Import | Export tab, and click on the **Import Config** button.



- Use the file browser to locate the .mtz files in ./artifacts/mtz.
- Select the securonix-entities.mtz package and click on **Next**.
- Use the checkboxes to select what you want to import and then click on **Next**.
- Complete the import
- Repeat steps 10-13 with the securonix-transforms.mtz package.

You should now be able to run the ThreatQ transforms on your Maltego entities.



Docker Shared Directories

The `./artifacts` directory is mounted as a shared volume between the host and container, when using Docker:

- Host path: `./artifacts`
- Container path: `/app/artifacts`
- Purpose: Persistent storage for generated artifacts (e.g., `.mtz` files, `.csv` files) that can be used in Maltego.

The `./config` directory is mounted as a shared volume between the host and container:

- Host path: `./config`
- Container path: `/app/config`
- Purpose: Persistent storage for configuration files, such as `threatquotient.cfg`, which contains the ThreatQ connection details and other settings. This file is optional and can be used instead of the `app.env` file for configuration.

Transforms

This integration comes with over 500 transforms. You'll be able to execute the following transforms on a variety of entity types:

- IOC -> Object Relationships
- Object -> Object Relationships
- STIX2 Object -> Object Relationships
 - Supported STIX2 Object Types: Attack Pattern, Course of Action, Threat Actor, Malware, Tool
- Fetch IOC Attributes as Properties
 - This includes attributes, tags, score, status, type, description, etc.
- Fetch Object Attributes as Properties
 - This includes attributes, tags, status, type, description, etc.

Upgrading the App

These steps will guide you through upgrading the ThreatQ TRX App to the latest version (after v2.1.0). If you are upgrading from a version prior to v2.1.0, please refer to the [Migrating from Previous Versions](#) section.

1. Make a copy of your existing `app.env` file (or `/config/threatquotient.cfg`) to preserve your settings.
2. Download the latest integration files from the ThreatQ Marketplace: <https://marketplace.threatq.com/details/maltego-trx-for-threatq>.
3. Extract the files to a directory on your server (or workstation).
4. Navigate into the extracted integration files directory in your terminal.
5. Run the following command to stop and remove the existing Docker container:

```
docker compose down -v --rmi all
```

This will stop the existing container and remove the associated volumes and images.

6. Copy your configurations from the previous `app.env` file (or `/config/threatquotient.cfg`) into the new `app.env` file (or `/config/threatquotient.cfg`) in the extracted directory.
 - Avoid replacing the entire file, as it may contain new settings that are required for the latest version.
7. Run the following command to build and start the new Docker container:

```
docker compose up -d
```

8. Follow the steps in the [Setting Up the TDS Instance](#) section to import the updated transforms into your TDS instance.
 - Or follow the steps (6+) in the [Installing Locally into Maltego Using Docker](#) section to import the updated transforms into your local Maltego client.

Migrating from Previous Versions

If you are upgrading from version 2.0.0 (or older) to version 2.1.0, you will need to perform the full installation steps again. This is due to the changes in how the transforms are packaged and run. The new version uses Docker to run the integration, which is a significant change from the previous version, which ran the app from the source code directly.

These changes were done to standardize the installation process as well as improve overall user-install experience. Now, python environments & dependencies are managed within the Docker container, and the transforms can be run on both TDS instances and local Maltego clients. In addition, installation on macOS vs. Linux vs. Windows is now consistent, as the Docker container abstracts away the differences in the underlying operating systems.



Ensure that you have backed up any configurations/settings (i.e. ThreatQ credentials) you may have set for the app prior to upgrading.

Development

Testing with a TDS Instance

If you do not have access to an TDS instance, you can still test the transforms using Maltego's Public TDS instance. This is useful for development and testing purposes. The public TDS instance mimics a standard TDS instance, acting as a proxy to your local transform server.

1. Register for an account on Maltego's Public TDS instance: <http://public-tds.paterva.com/>.
2. Install ngrok or a similar tool (i.e. `cloudflared`) to expose your local ThreatQ TRX app to the internet.

On macOS, you can install ngrok using Homebrew:

```
brew install ngrok
```

3. Add the ngrok authtoken generate on the ngrok website (run this command only once):

```
ngrok config add-authtoken <TOKEN>
```

4. Run ngrok to expose your local ThreatQ TRX app:

```
ngrok http 8080
```



This will provide you with a public URL that you can use to access your local ThreatQ TRX app. This URL should only be used for development purposes.

5. Make a copy of the `artifacts/csv/securonix-transforms.csv` file and modify the URL column to point to your ngrok URL (e.g. `https://<your-ngrok-url>/run/<transform>`).



You can do this in bulk by using a text editor's Find & Replace functionality, replacing `example.threatq.online` with your ngrok hostname.

6. Follow the steps in the [Installing into an TDS Instance](#) section to import the transforms into your TDS instance.



Use the modified `securonix-transforms.csv` file from step 5.

Packaging the TRX App for Distribution

To package the TRX app for distribution, you need to rebuild the package, generate new artifacts, and then compress the files into a tarball.

1. (One-Time): In your Python environment, install the build package:

```
pip install build
```

2. (Optional) Download the latest ThreatQ SDK .whl file and place it in the `dependencies` directory.
3. Run the packaging script which will install the package locally, generate updated artifacts, generate a new .whl file, and create a tarball for distribution:

```
# If you haven't yet, make the script executable
chmod +x package.sh
./package.sh
```



This will create a `threatq-maltego-trx-{VERSION}.tgz` file that can be distributed.

Known Issues / Limitations

- Maltego transforms that retrieve related domains (like IPv4 Address To Related Domains) automatically deduplicate domain variants. For instance, if ThreatQ returns both `www.amazon.co.jp` and `www.amazon.com`, the transform will only output the canonical Maltego entity, `amazon.com`.

Change Log

- **Version 2.1.0**
 - Added support for running transforms on a TDS instance.
 - Fixed STIX 2.1 object compatibility issues affecting certain transforms.
 - Standardized installation process for both TDS and local Maltego deployments.
 - Updated application to run in a Docker container using `docker-compose`.
 - Improved error handling for transform execution.
- **Version 2.0.0**
 - Complete rewrite of the integration based on the new Maltego TRX SDK.
 - Added support for running transforms on ThreatQ custom objects
 - Added transform to fetch object metadata (context: Attributes, Tags, Relationships, etc.)
 - Added transform to create an entity (i.e. Indicator, Malware, etc.) in ThreatQ.
 - Added support for running transforms on more entity types
- **Version 1.0.0**
 - Initial release