

ThreatQuotient



Resilient Connector for ThreatQuotient Implementation Guide

Version 1.1.0

Tuesday, May 26, 2020

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 © 2020 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.

Last Updated: Tuesday, May 26, 2020

Contents

Resilient Connector for ThreatQuotient Implementation Guide	1
Warning and Disclaimer	2
Contents	3
Introduction	4
Preface	4
Audience	4
Versioning	4
Assumptions	5
Installation	6
Installation Methods	6
From the ThreatQuotient Repository	6
Offline From the .Whl File	7
Add the Connector to ThreatQ UI	7
Cron	8
Configuration	10
Command Line Arguments	13
Generating a Certificate	14
Mapping	15

Introduction

Resilient Connector for ThreatQuotient allows new context from ThreatQ to be exported to your Resilient instance. It has the ability to push new indicators and comments from updated Resilient incidents in ThreatQ to Resilient as artifacts and comments, respectively.

Preface

This guide is to provide the information necessary to implement the Resilient Connector for ThreatQuotient. This document is not specifically intended to form a site reference guide.

It is assumed that the implementation engineer has experience installing and commissioning ThreatQuotient Apps and integrations covered within the document, as well as experience necessary to troubleshoot at a basic level.

Audience

This document is intended for use by the following parties:

1. ThreatQ Security/Engineers.
2. ThreatQuotient Professional Services Project Team & Engineers.

Versioning

- Current integration version: 1.1.0
- Supported on ThreatQ versions \geq 4.24

Assumptions

The following criteria is assumed to be in place and functional to allow the implementation of the Resilient Connector for ThreatQuotient into the managed estate:

- All ThreatQuotient equipment is online and in service.
- All required firewall ports have been opened.

Installation



If you are upgrading from a previous version of the integration, you will need to remove the current **Resilient.config** file on the ThreatQ instance before beginning the installation process listed below. The file will be located in the path specified by the user during the initial install of the connector. If a path was not specified during install, the configuration file will have been installed in the working directory at the time that the command was run.

This step is required as version 1.1.0 introduces new configuration fields. The new user fields will appear in the ThreatQ UI upon running the integration with the new configuration file.

You can install the Resilient Connector for ThreatQuotient using the following methods:

- [From the ThreatQuotient Repository](#)
- [Offline From the .Whl File](#)



The connector will need to be [added to the ThreatQ UI](#) and then [configured](#) before using.

Installation Methods

From the ThreatQuotient Repository

To install the Resilient Connector for ThreatQuotient integration from the ThreatQuotient repository with YUM credentials:

1. Install the Resilient Connector for ThreatQuotient connector using the following commands:

```
pip install tq-conn-resilient
```

Offline From the .Whl File

To install this Resilient Connector for ThreatQuotient from a wheel file, the wheel file (.whl) file will need to be copied via SCP into your ThreatQ instance.

1. Install the .whl file issuing the following command:

```
pip install tq_conn_resilient-<version>-py2-  
none-any.whl
```

Add the Connector to ThreatQ UI

After installing the connector, you will need to perform the following step in order for it to appear under the Incoming Feeds section of the ThreatQ platform. This will allow you to access the connector's configuration settings.

1. Run the following command to create a directory for the configuration and logs:

```
mkdir -p /opt/tq-integrations/resilient/  
  
tq-conn-resilient -v 3 -ll /opt/tq-  
integrations/resilient/ -c /opt/tq-  
integrations/resilient/
```

2. Run the following command manually start a run:

```
tq-conn-resilient -v 3 -ll stdout
```



The connector should be run on a schedule using crontab after it has been configured - See the [Cron](#) section for more details.

Cron

Use CRON or some other system schedule to run this script on a reoccurring basis . This can be run multiple times a day and can be run as often as required.



The argument in the cron script must specify the config and log locations.

Setting Up the CRONJOB

1. Log in via a CLI terminal session to your ThreatQ host.
2. Enter the command below.

```
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. If you have other scheduled commands, it is advised that you stagger the schedules.

3. Enter the following commands:



This example is for every 4 Hours.

CLI Crontab Command

```
0 */4 * * * tq-conn-resilient -v 3 -ll /opt/tq-  
integrations/resilient/ -c /opt/tq-  
integrations/resileint/
```


Configuration



ThreatQuotient does not issue API keys for third-party vendors. Contact the specific vendor to obtain API keys and other feed-related credentials.

To configure the connector :

1. Click on the **Settings** icon and select **Incoming Feeds**.
2. Locate the feeds under the **Labs** tab.
3. Click on the **Feed Settings** link for the connector.
4. Under the **Connection** tab, enter the following configuration parameters:

Parameter	Description
Resilient Host	The hostname or IP address of your Resilient instance.
Resilient Username	The email you will use authenticate with the Resilient API.
Resilient Password	The password you will use authenticate with the Resilient API.
Resilient Organization	Your Organization within your Resilient instance.
Resilient Certificate Path (Optional)	<div> The certificate must be accessible by the connector. The full path of the <code>.pem</code> file should be used in this field</div> <p>If this field is left blank, SSL will not be verified. See</p>


Parameter	Description
	the Generating a Certificate section to steps on how to create a certificate.
Attribute to Custom Field Mapping	<p>Maps ThreatQ attributes to Custom Fields in Resilient.</p> <p>Notes:</p> <ul style="list-style-type: none">• Each mapping must be on a new line• Each mapping is a equals-separated key/-value pair. <p>Example: Confidence=confidence_level</p> <ul style="list-style-type: none">• The Resilient Custom Field name must be the programmatic API name. See the Customization Settings in Resilient for more information.
ThreatQ Object to Custom Field Mapping	<p>Maps ThreatQ objects to Custom Fields in Resilient.</p> <p>Notes:</p> <ul style="list-style-type: none">• Each mapping must be on a new line• Each mapping is a equals-separated key/-value pair. <p>Example: TTP=mitre_technique_name</p> <ul style="list-style-type: none">• The Resilient Custom Field name must be the programmatic API name. See the Cus-

Parameter	Description
	tomization Settings in Resilient for more information.

5. Click on **Save Changes**.
6. Click on the toggle switch to the left of connector name to enable it.

Command Line Arguments

The Resilient Connector for ThreatQuotient supports the following custom command line arguments:

Argument	Details
<code>-hist, --historical [date]</code>	This argument will allow you to run a "historical" export. It will look for updated incidents after the supplied date.
<code>-n --name [connector name]</code>	<div>This argument allows you to install the connector with a custom name.</div> <div> This argument is mainly used when you want to install multiple instances of the connector. For instance, if you have multiple organizations or multiple instances of Resilient in your ecosystem.</div>

Generating a Certificate

You can generate a certificate for the connector, if needed, using the following command:

```
openssl s_client -connect <SERVER>:443 -showcerts -  
tls1 < /dev/null > cacerts.pem 2> /dev/null
```



The full path of the generated `.pem` file should be used as the certificate path in the connector configuration.

Mapping

The following is a list of context that is supported for sending updated information to Resilient.

ThreatQ	Resilient
Indicators	Artifacts
Malware Objects	Artifacts
Comments	Notes
Attributes	Custom Fields
Objects	Custom Fields