

ThreatQuotient



IronNet Connector Guide

Version 1.0.2

September 09, 2021

ThreatQuotient

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

Support

Email: support@threatq.com

Web: support.threatq.com

Phone: 703.574.9893

Contents

Versioning	4
Introduction	5
Prerequisites	6
Requirements.....	6
Time Zone Settings	6
Installation	7
Configuration	10
Usage	12
Command Line Arguments	12
Alerts Statuses	13
CRON.....	14
Endpoints	15
Authentication.....	15
Get Alerts	15
Set Alert Status	15
Average Connector Run	16
Change Log	17

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.

Versioning

- Current integration version: 1.0.2
- Supported on ThreatQ versions \geq 4.35.0
- Supported Python Versions: 2.7.X, 3.5.X

There are two versions of this integration:

- Python 2 version
- Python 3 version

Introduction

The IronNet connector ingests alerts and IoCs into ThreatQ as Events from an IronNet IronDefense appliance. The user can also change the status of the events in ThreatQ, with the change being synced back to the IronNet appliance. The connector pulls alerts based on a date and severity score range.

Prerequisites

The following items are required in order to use the integration.

Requirements

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.

Time Zone Settings

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
```

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_ironnet
```

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_ironnet  
  
pip download tq_conn_ironnet -d  
  
/tmp/tq_conn_ironnet/
```

- b. Archive the folder with the .whl files:

```
<> tar -czvf tq_conn_ironnet.tgz /tmp/tq_conn_ironnet/
```

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



A driver called `tq-conn-ironnet` will be installed. After installing with `pip` or `setup.py`, a script stub will appear in `/usr/bin/tq-conn-ironnet`.

- 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
```

- Perform an initial run using the following command:

```
<> tq-conn-ironnet -v3 -ll /var/log/tq_labs/ -c /etc/tq_labs/
```

- 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.
Email Address	This is the User in the ThreatQ System for integrations.
Password	The password for the above ThreatQ account.
Status	This is the default status for objects that are created by this Integration.

PARAMETER

DESCRIPTION

It is common to set this field to Active but your organization SOPs should be respected when setting this field.

Example Output

```
tq-conn-ironnet -v3 -ll /var/log/tq_labs/ -c /etc/tq_labs/  
ThreatQ Host: <ThreatQ Host IP or Hostname>  
ThreatQ Client ID: <ClientID>  
E-Mail Address: <EMAIL ADDRESS>  
Password: <PASSWORD>  
Status: Active  
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
IronNet Hostname	The hostname or IP address for your IronNet instance.
Port	Enter the port number to access. The default is 443.
IronNet Username	The username for your IronNet account.
IronNet Password	The password for your IronNet account.
Severity Lower Bound	The lower severity score bound. The default is 500.
Severity Upper Bound	The Upper severity score bound. The default is 1000.

IronNet

Feed Settings

Connection

Settings

API IP/Hostname

https://aws-dev.irondev.io

Hostname or IP address of the IronNet instance.

Port. Default is 443

443

Port for communicating with the IronNet API.

Username

threatq

Username for logging to IronNet.

Password

.....

Password for logging to IronNet.

severity lower bound

250

Enter the lower bound for the alert severity.

severity upper bound

500

Enter the upper bound for the alert severity.

initial_query

1

Enter how many days worth of data you would like to pull for the initial run.

Save Changes

- IronNet Connector
Version 1.0.2

Usage

Use the following command to execute the driver:

```
<> tq-conn-ironnet -v3 -ll /var/log/tq_labs/ -c /etc/tq_labs/
```

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>-ll LOGLOCATION, --loglocation LOGLOCATION</code>	Sets the logging location for the connector. The location should exist and be writable by the current.
<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.)
<code>-v {1,2,3}, --verbosity {1,2,3}</code>	This is the logging verbosity level where 3 means everything.
<code>-ep, --external-proxy</code>	This allows you to use the proxy that is specified in the ThreatQ UI. This specifies an internet facing proxy, NOT a proxy to the TQ instance.

ARGUMENT	DESCRIPTION
<code>-s, --start-historical</code>	The start date for a user defined query in YYYY-MM-DDTHH:MM:SS format. If not provided the connector will set the start date to the last successful run time.
<code>-e, --end-historical</code>	The start date for a user defined query in YYYY-MM-DDTHH:MM:SS format. If not provided the connector will set the start date to the current time.



All location-based options default to the current working directory if they are not provided. To find additional options and option descriptions, invoke the program with `-h`.

Alerts Statuses

The connector can change statuses of alerts on the IronNet appliance via custom event types in ThreatQ:

EVENT TYPE	IRONNET ALERT STATUS
CLOSED	STATUS_CLOSED
DISCARDED	STATUS_CLOSED
UNDER REVIEW	STATUS_UNDER_REVIEW
AWAITING REVIEW	STATUS_AWAITING_REVIEW

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 hour.

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 Hour Example

```
<> 0 * * * * tq-conn-ironnet -c /etc/tq_labs/ -ll /var/log/
    tq_labs/ -v3
```

4. Save and exit CRON.

Endpoints

The following endpoints are used with the connector.

Authentication

`https://<IronNet Host>/IronApi/Login`

Authentication Purpose: Obtain an authentication token from the IronNet instance.

Get Alerts

`https://<IronNet Host>/IronApi/GetAlerts`

Get Alerts Purpose: Query the IronNet instance for alerts to ingest.

Set Alert Status

`https://<IronNet Host>/IronApi/SetAlertStatus`

Set Alert Status Purpose: Close or open alerts in IronNet based on their event type in ThreatQ.

Average Connector Run



Object counts and Feed runtime are supplied as generalities only - objects returned by a provider can differ based on credential configurations and Feed runtime may vary based on system resources and load.

EXPORTED OBJECT COUNT	FEED RUNTIME
-----------------------	--------------

500	10 minutes
-----	------------

1,000	20 minutes
-------	------------

10,000	200 minutes
--------	-------------

Change Log

- **Version 1.0.2**
 - Fixed a bug regarding URL scheme and port.
- **Version 1.0.1**
 - Addressed a dateutil dependency bug.
- **Version 1.0.0**
 - Initial Release