Expanse Tenable.sc Integration v1.0.0

Documentation and User Guide

Expanse's Tenable.sc integration allows you to automatically import assets detected by Expanse into your Tenable.sc instance. This documentation details the requirements for the integration, how to set up and run the integration, how to configure the integration, and how to debug common issues.

Goals and Outcomes

Examples of goals and outcomes for customers using the Expanse Integration include:

- Keeping an up-to-date inventory in Tenable.sc of an organization's external facing assets by using data from Expander.

Integration Requirements

Recommended System Requirements

To run the integration, you will need a system on which you can leave a continuously running Docker container. Check the Docker installation guide for the system requirements to install Docker. In addition, we recommend you have at least 1 GB of RAM on top of what is required for Docker to run the application.

Docker

You will need to install Docker to run the integration. Check the Docker installation guide for guidance on the installation process.

Python

To run the integration in ad-hoc mode you can just run the python script directly. This requires python 3.7+.

Network Requirements

The system on which the integration is running will need to have a network connection through which it can access both https://expander.expanse.co/ and your Tenable.sc instance. The integration communicates to both Expander and Tenable.sc over HTTPS, connecting to port 443 for Expanse and whatever port is necessary to communicate with your Tenable.sc instance.

Tenable.sc Permissions

In order for the integration to upload asset data to Tenable.sc you will need to grant it the necessary permissions in Tenable.sc to read and write new asset lists.

Setting up and Running the Integration

Setup

1. Ensure that Docker is installed by running the command "`docker version`". If Docker is not installed, you can follow the directions here to install it: https://docs.docker.com/engine/install/

2. Unpack the tarball.

3. Edit the "expanse_tenable.yml" file in the root directory of the application to set your desired configuration settings. Reference the "Configuring Your Expanse Tenable.sc Integration" section below for more details.

4. Open the "crontab" file in a text editor. On the second to last line, you will see "* * * * *" followed by a series of commands. Edit the "* * * * *" to adjust the run schedule to your desired frequency
   NOTE: Docker uses Greenwich Mean Time by default, so you will need to set your run schedule in GMT. For reference, GMT is 4 hours ahead of EST, and 7 hours ahead of PST.
   Some common examples:

Schedule	Cron Schedule Expression
Every day at midnight GMT (8pm EST / 5pm PST)	0 0 * * *
Every day at midnight EST (4am GMT / 9pm PST)	0 4 * * *
Every day at midnight PST (3am EST / 7am GMT)	0 7 * * *
Every Sunday at midnight PST	0 7 * * 0
Every Friday at midnight PST	0 7 * * 5

   See https://crontab.guru/examples.html for more examples on how to do this. Keep in mind that Docker uses Greenwich Mean Time by default, so you will need to set your run schedule in GMT. For reference, GMT is 4 hours ahead of EST, and 7 hours ahead of PST.

Running

If you do not intend on using any environment variables you can run the command "`sh run.sh`" from the root project directory to create and run the docker image. This will begin the cron job, and the Python script will run according to the frequency in crontab within a Docker container

If you plan on using environment variables you can run the docker command manually with your environment variable values.

```
docker build -t expanse_tenablesc . && docker run -it --rm -e
EXPANSE_TENABLE_SC_USERNAME=$USERNAME -e
EXPANSE_TENABLE_SC_PASSWORD=$PASSWORD expanse_tenablesc
```

To force execution of a single run of the Python script, run "`python expanse_tenablesc`" from the root project directory.

Configuring Your Expanse Tenable.sc Integration

Configuration is handled through the expanse_tenable.yml file, and optionally, through environmental variables. Below is an example of what the expanse_tenable.yml file should look like after being filled out. In this example we are importing all assets from Expanse with the tag "new acquisition" and adding these to two new asset lists.

```
tenable:
  username: fake_username

  password: fake_password

  url: tenable.company.com

  dns_asset_list_name: Expanse New Acquisition Domains

  ip_asset_list_name: Expanse New Acquisition IPs

expanse:
  api_token: dsFsAjw3f_n36712wuhed-adetweb44ARgdtt

  tags:
    - new acquisition

  responsive_only: False

general:
  logging_level: INFO

  logging_location: expanse_tenable.log
```

Tenable.sc Config

tenable.username

The username to use when making API calls to Tenable.sc. Can also be passed in as the environment variable EXPANSE_TENABLE_SC_USERNAME.

tenable.password

The password to use when making API calls to Tenable.sc. Can also be passed in as the environment variable EXPANSE_TENABLE_SC_PASSWORD.

tenable.access_key

API Access Key for Tenable.sc used to authenticate requests. Required if tenable.username / tenable.password are not supplied. Can also be passed in as the environment variable EXPANSE_TENABLE_SC_ACCESS_KEY.

tenable.secret_key

API Secret Key for Tenable.sc used to authenticate requests. Required if tenable.username / tenable.password are not supplied. Can also be passed in as the environment variable EXPANSE_TENABLE_SC_SECRET_KEY.

tenable.url

The URL of your Tenable.sc instance. Should not include protocol or port. Ex. tenable.company.com Can also be passed in as the environment variable EXPANSE_TENABLE_SC_URL.

tenable.port

The port used for https communication to Tenable.sc instance. If this is 443 it can be omitted.

tenable.dns_asset_list_name

The name that should be used for any DNS Name based asset list in Tenable.sc. If this variable is specified the integration WILL collect domains from Expanse, otherwise if left blank it WILL NOT import domains into Tenable.sc.

tenable.ip_asset_list_name

The name that should be used for any Static IP based asset list in Tenable.sc. If this variable is specified the integration WILL collect IP Ranges from Expanse, otherwise if left blank it WILL NOT import IP Ranges into Tenable.sc.

Expanse Config

expanse.api_token

The refresh token used to get data from Expander. Can also be passed in as the environment variable EXPANSE_API_TOKEN.

expanse.tags

This filter can be used to specify which assets you'd link to import based on their tag name. If no value is configured then no tag filtering will occur. These should be included in the config file as a yaml sequence.

expanse.business_units

This filter can be used to specify which assets you'd link to import based on their Business Unit name. If no value is configured then no Business Unit filtering will occur. These should be included in the config file as a yaml sequence.

expanse.responsive_only

A flag that will cause only responsive assets to be imported into Tenable.sc. If True, only responsive assets will be imported, if False all assets will be imported. This is set to False by default.

General Config

general.logging_level

The level of detail of logs. Should be either DEBUG, INFO, WARNING, or ERROR. Default is INFO.
All logs of equal or greater severity to the selected level will be shown. Below are descriptions of each level in increasing order of severity.

DEBUG	Detailed info about all HTTP requests and actions taken.
INFO	General info giving an overview of the program's progress.
WARNING	Info about abnormal events that do not immediately present a problem, but may cause issues later.
ERROR	Info about problems that prevent the program from performing a required action.

general.logging_location

By default the integration will log to stdout as well as to file. This sets the logging location.

Environment Variables

To use environment variables for certain config settings, simply make sure that when you run the Docker container, it is running with the correct environment variables. This can be accomplished, for instance, by using the "-e" tag in the "docker run" command.

For example, to set the EXPANSE_TENABLE_SC_USERNAME and EXPANSE_TENABLE_SC_PASSWORD environment variables to "myUser" and "myPassword" respectively, one could use the following command:

```
docker build -t expanse_tenablesc . && docker run -it --rm -e EXPANSE_TENABLE_SC_USERNAME=myUser -e
EXPANSE_TENABLE_SC_PASSWORD=myPassword expanse_tenablesc
```

Troubleshooting the Integration

If you encounter issues with the integration you are encouraged to enable debug logging to facilitate faster diagnosis of issues. You can do this by setting **general.logging_level** to **DEBUG** in the YAML config.