Expanse Venafi Integration v1.1.0

Documentation and User Guide

Expanse's Venafi Integration is a multi use script that allows users to audit their Venafi certificate inventory by comparing it with Expanse's collection of externally facing certificates. The script also allows users to import missing certificates that have been discovered by Expanse into their Venafi TPP instance. Either of these actions may be configured to filter Expanse certificates by asset tag or their attributed business unit.

Goals and Outcomes

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

- Audit current Venafi Certificate inventory.
- Import newly discovered certificates from Expanse into Venafi.

Overview of Functionality

Overview

The Expanse Venafi Integration has two modes, **audit** and **import**.

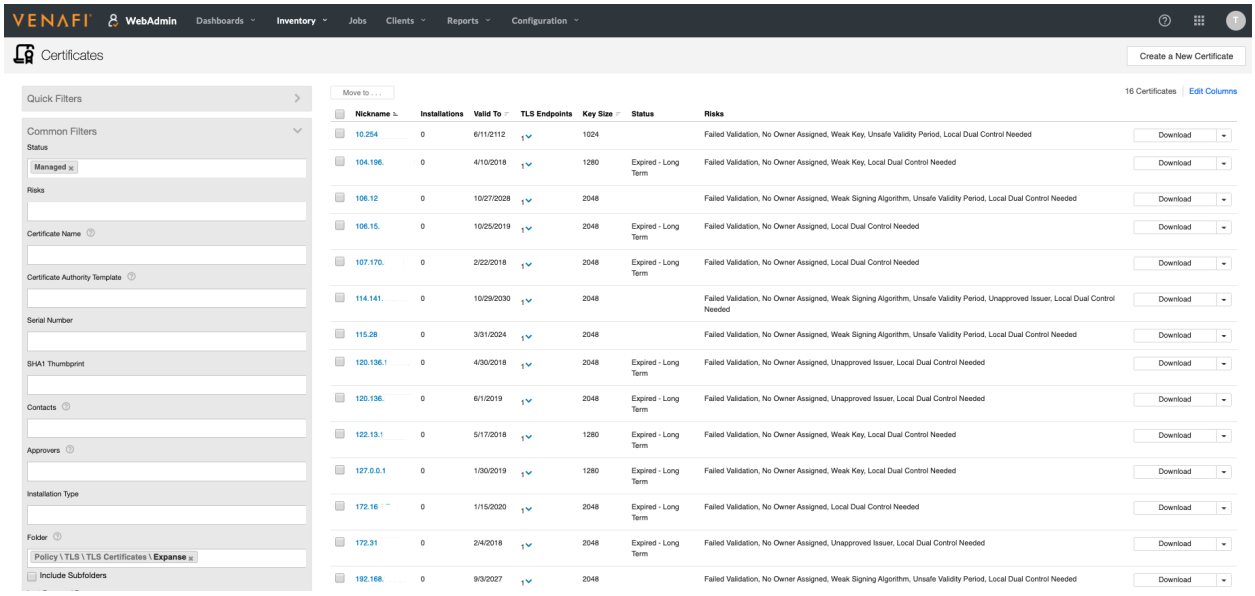
Audit

By default it is run in **audit** mode meaning it will produce a delta as a .csv file and will not make any write requests to your Venafi instance.

Import

In **import** mode, which can be configured by supplying the **--import-to-venafi** flag at run time, the integration will produce a delta as well as import all certificates into a targeted Venafi Policy that exist in Expanse and do not Exist in Venafi.

Results



Imported certificates

The generated .csv file will contain the following columns.

Column Name	**Meaning**	**Example**
Common Name	The certificate Common Name.	company.example.com
Venafi Serial	The serial number according to Venafi. This is in Hex value.	0606D97F8028DD681C2566C88583E366
Venafi Thumbprint	The Venafi certificate thumbprint. This is a Hex value.	D085D49233AF813A581E9B8221BAA70BF5CD9769
Venafi ValidTo	The Venafi certificate ValidTo date.	2019-04-30T00:00:00.0000000Z
Venafi ValidFrom	The Venafi certificate ValidFrom date.	2020-04-01T12:00:00.0000000Z
Venafi Issuer	The Venafi certificate Issuer.	CN=COMODO RSA Organization Validation Secure Server CA, O=COMODO CA Limited, L=Salford, S=Greater Manchester, C=GB
Expanse Serial	The serial number according to Expanse. This is a decimal value.	16187451376937929413

Expanse Fingerprint	The Expanse certificate fingerprint. This is a base64 encoded hash.	4u7csDeRHKSqf_hMecfOfIXX5Ks=
Expanse ValidFrom	The Expanse certificate ValidFrom date.	2011-10-28T05:25:50Z
Expanse ValidTo	The Expanse certificate ValidTo date.	2021-10-25T05:25:50Z
Expanse Issuer	The Expanse certificate Issuer.	C=GB,S=Greater Manchester,L=Salford,O=Fake COMODO CA Limited,CN=COMODO RSA Organization Validation Secure Server CA
Discovered	This field reflects where the certificate was discovered. Values can be EXPANSE, VENAFI, or BOTH.	EXPANSE

Installing the Integration

The integration is delivered as a Python package in compressed tar.gz format.

Requirements

- Linux/Unix Host with a minimum of 2 GB RAM
- Python 3.7+
- An Internet connection for outgoing requests and access to your Venafi TPP instance.

Installing Python Dependencies

You can use pip to install python dependencies. Within the base directory of the integration there is a file named requirements.txt which can be used with pip to install the necessary python modules. It is also recommended that you use a virtual environment to ensure you don't encounter any global dependency clashes.

To install run:

```
pip install -r requirements.txt
```

Configuring the Integration

The Expanse Venafi Integration configuration is quite configurable in order to achieve the desired results. Configuration of the integration is primarily done using a YAML file named **expanse.yml**.

You should find a file named **EXAMPLE_expanse.yml** which can be copied and renamed to **expanse.yml** and will serve as a good starting point for configuration.

YAML Configuration

Field Name	Description	Required	Default Value
venafi.username	Username in Venafi TPP.	Yes	N/A
venafi.password	Password for user in Venafi TPP.	Yes	N/A
venafi.client_id	This is the ID for the API Application Integration described below.	Yes	N/A
venafi.url	This is the base url for your Venafi TPP instance.	Yes	N/A
venafi.base_dn	This is the DN that your target policy belongs to. ex. `**\\VED\\Policy\\TLS\\TLS Certificates**`	Yes	N/A
venafi.target_policy	This is the policy name that you'd like all Expanse certificates to be imported into.	Yes	N/A
expanse.api_token	API token for Expanse. This is used to authenticate requests to the Expanse API.	Yes	N/A
expanse.tags	This filter can be used to specify which certificates from Expanse should be pulled for comparison or Import. Only certificates that have been assigned the included tags will be returned.	No	N/A

expanse.business_units	This filter can be used to specify which certificates from Expanse should be pulled for comparison or Import. Only certificates belonging to one of the Business Units included will be returned.	No	N/A
general.logging_level	The logging level for the integration. Valid options are DEBUG, INFO, WARN, and ERROR	No	INFO
general.logging_location	By default the integration will log to stdout as well as to file. This sets the logging location.	Yes	N/A
general.compare_issuer	Whether or not the integration should compare Issuer in addition to Serial Number when matching certificates. This provides greater accuracy, but reduces performance.	No	False

Example

This example would be for a customer who wanted to import only certificates with the tag, 'verified'.

```
venafi:
  username: local:tppadmin
  password: Password
  client_id: Expanse
  url: https://venafitpp.company.com
  base_dn: \\VED\\Policy\\TLS\\TLS Certificates
  target_policy: Expanse

expanse:
  api_token: 12345678987654dfgvhb59jfeu24rjiwrf
  tags:
     - verified

general:
  logging_level: INFO
  logging_location: expanse_venafi.log
```

Environment Variables

For some of the more sensitive values, environment variables can also be used in addition to the YAML config file. If environment variables are provided they will override any values provided in the YAML config file.

Name	**Corresponding YAML Field**	**Description**

EXPANSE_VENAFI_PASSWORD	venafi.password	Password for user in Venafi TPP.
EXPANSE_VENAFI_USERNAME	venafi.username	Username in Venafi TPP.
EXPANSE_VENAFI_URL	venafi.url	This is the base url for your Venafi TPP instance.
EXPANSE_VENAFI_CLIENT_ID	venafi.client_id	This is the ID for the API Application Integration described below.
EXPANSE_API_TOKEN	expanse.api_token	API token for Expanse. This is used to authenticate requests to the Expanse API.

Command Line Arguments

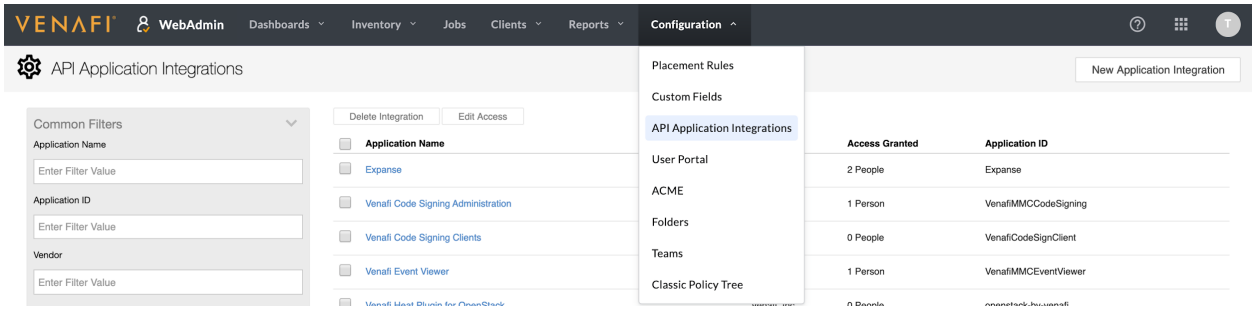
In addition to the YAML config file and environment variable, there are also some runtime configurations that can be made via command line arguments.

Arg Name	Description	Required	Default
--conf	The location of the YAML config file.	No	./expanse.yml
--import-to-venafi	Runs the tool in `**import**` mode.	No	false

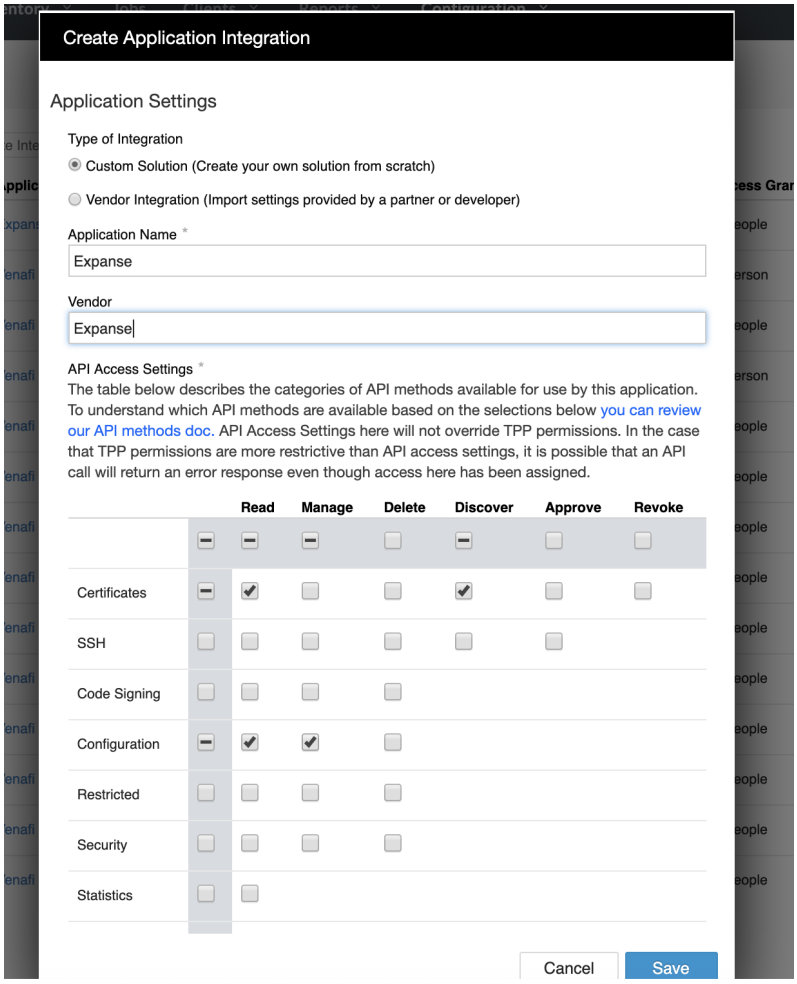
Creating a new API Application Integration in Venafi

In order to run the integration you will need to create a new API Application Integration in Venafi. The integration requires `**Discover**` permissions for `**Certificates**` and `**Manage**` permissions for `**Configuration**`.

You can set up a new API Application Integration through the Venafi UI in the top menu `Configuration > API Application Integrations`.



Finding API Application Integrations through WebAdmin



Creating a new API Application Integration

You can also reference Venafi's documentation for further details by navigating to `/aperture/help/Content/API-ApplicationIntegration/c-APIAppIntegrations-about.htm?Highlight=API%20Application` on your own instance of Venafi TPP.

Running the Integration

Running with Python

1. [Optional] Create a new virtual environment
2. Install python dependencies using pip.
3. Update the PYTHONPATH environment variable to include the path of the integration.

```
export PYTHONPATH=$PYTHONPATH:$(pwd)
```

4. Create a new file in the base of the directory of the integration named **expanse.yml**. Alternatively you can actually place this file in any directory you want with any name you want if you intend to use --conf command line argument. Configure this file based on the options and examples above.
5. [OPTIONAL] configure any environment variables based on the options above.
6. Start the integration by running

```
python src
```

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.