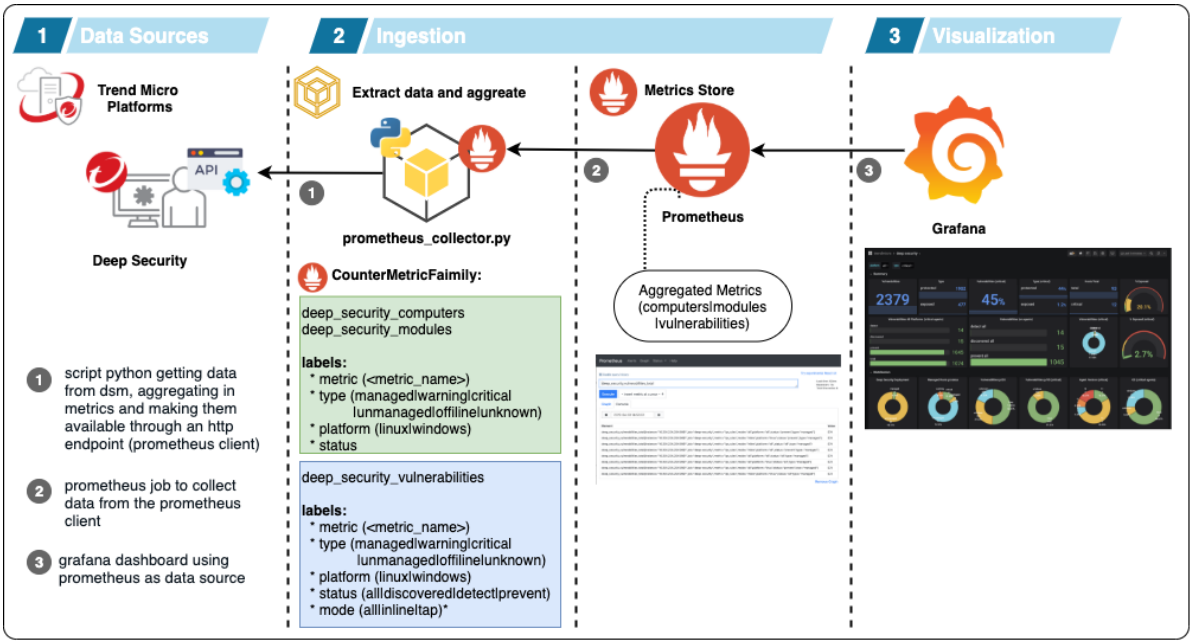
tmds11-exporter

This project creates a prometheus collector getting metrics from Deep Security DSM 11.0.

The data is aggregated in count and segmented in 3 groups:

- deep_security_computers
- deep_security_modules
- deep_security_vulnerabilities



prometheus labels

- **deep_security_computers**
  - labels:
    - metric: *platform* | *os_type* | *agent_version* | *agent_version_major*
    - type: *managed* | *warning* | *critical* | *unmanaged* | *offline* | *unknown*
    - platform: *all* | *linux* | *windows*
    - status: *(os version)* | *(agent version)*
- **deep_security_modules**
  - labels:
    - metric: *am_status* | *wr_status* | *fw_status* | *ip_status* | *im_status* | *li_status*
    - type: *managed* | *warning* | *critical* | *unmanaged* | *offline* | *unknown*
    - platform: *all* | *linux* | *windows*
    - status: *on* | *off*
- **deep_security_vulnerabilities**

    - labels:
        - metric: *am_status* | *wr_status* | *fw_status* | *ip_status* | *im_status* | *li_status*
        - type: *managed* | *warning* | *critical* | *unmanaged* | *offline* | *unknown*
        - platform: *linux* | *windows*
        - status: *all* | *discovered* | *detect* | *prevent*

About vulnerabilities status:

- **discovered:** vulnerabilities that are detected but the IPS is not enabled on the host
- **detect:** vulnerabilities with IPS enabled but configured on detect mode
- **prevent:** vulnerabilities with IPS enabled and configured on prevent mode

environment:

- **python**: *python 2.7* (required)
- **prometheus**: *v2.16* (tested with this version)
- **grafana**: *6.6.2* (tested with this version)

configuration

create a virtual environment

virtualenv

```
virtualenv venv
source venv/bin/activate
pip install -r requirements.txt
```

pipenv

```
pipenv --two
pipenv shell
pip install -r requiriments.txt
```

running the app:

You should configure a config.py (**renaming config_sample.py to config.py** with your configuration), or using environment variables, to configure:

Variable	Description	Value	Value Type
DS_HOST	DSM Hostname	ip	fqdn
DS_PORT	DSM TCP Port	port Number	string
DS_USER	User Account (read only)	user_name - base64 encoded	string

Variable	Description	Value	Value Type
DS_PASS	User Password	user_pass - base64 encoded	string
DS_VERIFY_SSL	SSL Verify	True	False
DS_API_CHECK	Cache API data	time in minutes	integer
SERVER_PORT	Prometheus Collector TCP Port	port number	integer
LOG_LEVEL	Log level	INFO	WARN

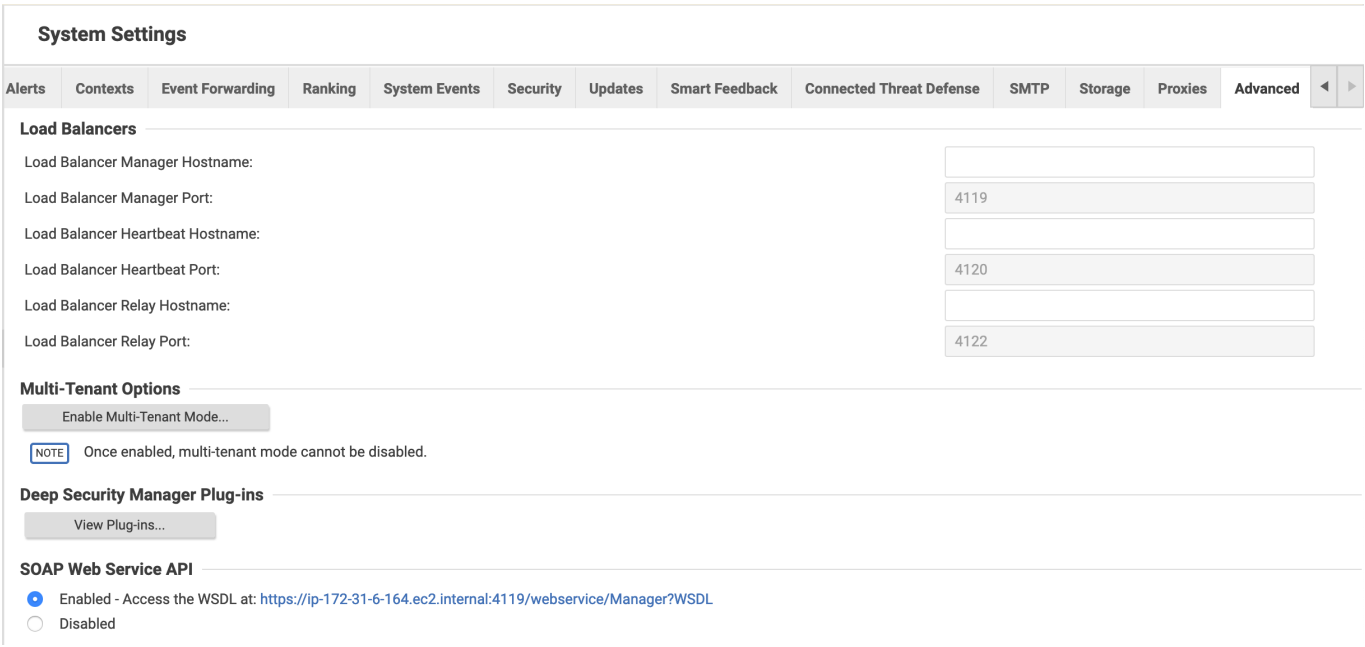
To encode your credentials:

```
echo -ne '<ds_user>'|base64
echo -ne '<ds_pass>'|base64
```

enabling soap web api

We need to enable SOAP Web API on the DSM. To do it, you should to to:

- *Administration* tab
- - System settings* pane
- *SOAP Web Service API* option - check '**enable**' radio button



grafana dashboard:

Import the dashboard located on: grafana/dash.json

- **dashboard:**

- **filtering by type:**