PyMISP - (ab)using MISP API with PyMISP and Viper

MISP - Malware Information Sharing Platform & Threat Sharing

CIRCL
Computer Incident
Response Center
Luxembourg

Raphaël Vinot - *TLP:WHITE*

June 16, 2016

PyMISP - Basics

- Installation:
  - pip install pymisp
- Get your auth key from:
  - https://misppriv.circl.lu/events/automation
- Fetch the repository to get the examples:
  - git clone https://github.com/MISP/PyMISP.git

PyMISP - Examples

- Usage:
  - Create examples/keys.py with the following content

```
misp_url = "https://mispppriv.circl.lu"
misp_key = "<API_KEY>"
misp_verifycert = True
```

- **PyMISP needs to be installed**

PyMISP - Examples

- All the examples have help if you do **script.py -h**
- **searchall.py**: Search in the whole database for a value
- **last.py**: Returns all the most recent events (on a timeframe)
- **get.py**: Return a specific event
- **yara.py**: Get Yara rules
- **suricata.py**: Get Suricata rules
- **tags.py**: Returns all the tags activted on the platform
- **get_network_activity.py**: Returns network indicators

PyMISP - Examples

- **copy_list.py**: Copy files from one MISP instance to an other
- **create_events.py**: Create an event
- **up.py**: Update an event
- **upload.py**: Upload a malware sample
- **sighting.py**: Update sightings on an attribute
- **stats.py**: Returns the stats of a MISP instance

PyMISP - Usage

- Basic example

```
from pymisp import PyMISP
api = PyMISP(url, apikey, verifycert=True, 'json', debug=False)
response = api.<function>
if response['error']:
    <something went wrong>
else:
    <do something with the output>
```

PyMISP - Capabilities

- Events: get, add, update, publish and delete
- Events, more: change Threat level, add tag
- Add file attributes: hashes, registry key, patterns, pipe, mutex
- **Update sightings**
- Add network attributes: IP dest/src, hostname, domain, url, UA, ...
- Add Email attributes: source, destination, subject, attachment, ...
- Upload/download samples
- Proposals: add, edit, accept, discard
- **Full text search** and search by attributes
- Tags: get and create
- Get API and platform version
- And more, look at the api file

PyMISP - Feed generator

- Used to generate the **CIRCL OSINT feed**
- Export events as json based on tags, organisation, events, ...
- Automatically update the dumps and the metadata file
- Comparable to a lighweight **TAXII interface**

PyMISP - Feed generator - Config file

```
url = ''

key = ''

ssl = True

outputdir = 'output'

# filters = {'tag':'tlp:white|feed-export|!privint','org':'CIRCL'}
filters = {}


valid_attribute_distribution_levels = ['0', '1', '2', '3', '4', '5']
```

PyMISP - OpenIOC to MISP

- Easy **import of OpenIOC** files into MISP
- Possible to set specific tags
- Batch import

Viper

- **Solid CLI**
- Django interface is available (I've been told)
- Plenty of modules
- Locale storage of your own zoo

PyMISP & Viper

- Full featured **CLI for MISP**
- Search / **Cross check with VT**
- Create / Update / Show / Publish Event
- Download / Upload Samples
- Mass export / upload / download
- Get Yara rules

Viper & VT

- Searches for hashes/ips/domains/URLs from the current MISP event, or download the samples
- Download samples from current MISP event
- Download all samples from all the MISP events of the current session

Other modules

- Fully featured CLI for **Passive SSL**
- Fully featured CLI for **Passive DNS**
- Can launch Radare2 or IDA
- ... And let's look at it in the demo.

Q&A

- `https://github.com/MISP/PyMISP`
- `https://github.com/MISP/`
- `https://github.com/viper-framework/viper`
- We welcome new functionalities and pull requests.