EXTENDING MISP WITH PYTHON MODULES

MISP - Threat Sharing

CIRCL / Team MISP Project

HTTP://WWW.MISP-PROJECT.ORG/
TWITTER: @MISPPROJECT

NSPA
Ways to extend MISP before modules

- APIs (PyMISP, MISP API)
  - Works really well
  - No integration with the UI
- Change the core code
  - Have to change the core of MISP, diverge from upstream
  - Needs a deep understanding of MISP internals
  - Let’s not beat around the bush: Everyone hates PHP
G O A L S F O R T H E M O D U L E S Y S T E M

- Have a way to extend MISP without altering the core
- Get started quickly without a need to study the internals
- Make the modules as light weight as possible
  - Module developers should only have to worry about the data transformation
  - Modules should have a simple and clean skeleton
- In a friendlier language - Python
Extending MISP with expansion modules with zero customization in MISP.

A simple ReST API between the modules and MISP allowing auto-discovery of new modules with their features.

Benefit from existing Python modules in Viper or any other tools.

MISP modules functionnality introduced in MISP 2.4.28.

MISP import/export modules introduced in MISP 2.4.50.
MISP modules can be run on the same system or on a remote server.

Python 3 is required to run MISP modules.

- `sudo apt-get install python3-dev python3-pip libpq5`
- `cd /usr/local/src/`
- `sudo git clone https://github.com/MISP/misp-modules.git`
- `cd misp-modules`
- `sudo pip3 install -I -r REQUIREMENTS`
- `sudo pip3 install -I .`
- `sudo vi /etc/rc.local, add this line: ‘sudo -u www-data misp-modules -s &’`
MISP modules - Simple REST API mechanism

- http://127.0.0.1:6666/modules - introspection interface to get all modules available
  ▶ returns a JSON with a description of each module
- http://127.0.0.1:6666/query - interface to query a specific module
  ▶ to send a JSON to query the module

- MISP autodiscovers the available modules and the MISP site administrator can enable modules as they wish.

- If a configuration is required for a module, MISP adds automatically the option in the server settings.
curl -s http://127.0.0.1:6666/modules

```json
{
  "type": "expansion",
  "name": "dns",
  "meta": {
    "module-type": [
      "expansion",
      "hover"
    ],
    "description": "Simple DNS expansion service to resolve IP address from MISP attributes",
    "author": "Alexandre Dulaunoy",
    "version": "0.1"
  },
  "mispattributes": {
    "output": [
      "ip-src",
      "ip-dst"
    ],
    "input": [
      "hostname",
      "domain"
    ]
  }
}
```
## MISP MODULES - CONFIGURATION IN THE UI

### Server settings

<table>
<thead>
<tr>
<th>Priority</th>
<th>Setting</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical</td>
<td>Plugin.Enrichment_services_enable</td>
<td>true</td>
<td>Enable/disable the enrichment services.</td>
</tr>
<tr>
<td>Recommended</td>
<td>Plugin.Enrichment_services_url</td>
<td><a href="http://127.0.0.1">http://127.0.0.1</a></td>
<td>The url used to access the services.</td>
</tr>
<tr>
<td>Recommended</td>
<td>Plugin.Enrichment_services_port</td>
<td>6666</td>
<td>The port used to access the services.</td>
</tr>
<tr>
<td>Recommended</td>
<td>Plugin.Enrichment_cve_enabled</td>
<td>false</td>
<td>Enable or disable the cve enabled setting.</td>
</tr>
<tr>
<td>Recommended</td>
<td>Plugin.Enrichment_dns_enabled</td>
<td>true</td>
<td>Enable or disable the dns metrics.</td>
</tr>
<tr>
<td>Recommended</td>
<td>Plugin.Enrichment_sourcecache_enabled</td>
<td>false</td>
<td>Enable or disable the sourcecache enabled setting.</td>
</tr>
<tr>
<td>Recommended</td>
<td>Plugin.Enrichment_sourcecache_archivepath</td>
<td></td>
<td>Set this required module specific setting.</td>
</tr>
<tr>
<td>Recommended</td>
<td>Plugin.Enrichment_passivetotal_enabled</td>
<td>true</td>
<td>Enable or disable the passivetotal enabled setting.</td>
</tr>
<tr>
<td>Recommended</td>
<td>Plugin.Enrichment_passivetotal_username</td>
<td><a href="mailto:alexandre.dulaunoy@circl.lu">alexandre.dulaunoy@circl.lu</a></td>
<td>Set this required module specific setting.</td>
</tr>
<tr>
<td>Recommended</td>
<td>Plugin.Enrichment_passivetotal_password</td>
<td></td>
<td>Set this required module specific setting.</td>
</tr>
</tbody>
</table>
MISP modules - How it’s integrated in the UI?

Choose the enrichment module that you wish to use for the expansion

dns

Enrichment Results

Below you can see the attributes that are to be created. Make sure that the categories and the types are correct; often several options will be offered based on an inconclusive automatic resolution.
Expansion modules - enrich data that is in MISP
▶ Hover type - showing the expanded values directly on the attributes
▶ Expansion type - showing and adding the expanded values via a proposal form

Import modules - import new data into MISP

Export modules - export existing data from MISP
curl -s http://127.0.0.1:6666/query -H "Content-Type: application/json" –data @body.json -X POST

body.json

```
{
  "module": "dns",
  "hostname": "www.circl.lu"
}
```

and the response of the dns module:

```
{
  "results": [{
    "values": ["149.13.33.14"],
    "types": ["ip-src", "ip-dst"]
  }]
}
```
import json
import dns.resolver
misperrors = {'error': 'Error'}
mispattributes = {'input': ['hostname', 'domain'], 'output': ['ip-src', 'ip-dst']}
moduleinfo = {'version': '0.1', 'author': 'Alexandre Dulaunoy',
'description': 'Simple DNS expansion service to resolve IP address from MISP attributes', 'module-type': ['expansion', 'hover']}
def handler(q=False):
    if q is False:
        return False
    request = json.loads(q)
    if request.get('hostname'):
        toquery = request['hostname']
    elif request.get('domain'):
        toquery = request['domain']
    else:
        return False
    r = dns.resolver.Resolver()
    r.timeout = 2
    r.lifetime = 2
    r.nameservers = ['8.8.8.8']
    try:
        answer = r.query(toquery, 'A')
    except dns.resolver.NXDOMAIN:
        mispattributes['error'] = 'NXDOMAIN'
        return mispattributes
    except dns.exception.Timeout:
        mispattributes['error'] = 'Timeout'
        return mispattributes
    except:
        mispattributes['error'] = 'DNS resolving error'
        return mispattributes
    r = [{'results': [{'types': mispattributes['output'], 'values': [str(answer[0])]}]}
return r

def introspection():
    return mispattributes

def version():
    return moduleinfo
Testing your module

- Copy your module dns.py in modules/expansion/
- Restart the server misp-modules.py
  ```
  $ python3 misp-modules.py
  2016-03-20 19:25:43,748  - misp-modules - INFO - MISP modules passivetotal imported
  2016-03-20 19:25:43,787  - misp-modules - INFO - MISP modules sourcecache imported
  2016-03-20 19:25:43,789  - misp-modules - INFO - MISP modules cve imported
  2016-03-20 19:25:43,790  - misp-modules - INFO - MISP modules dns imported
  2016-03-20 19:25:43,797  - misp-modules - INFO - MISP modules server started on TCP port 6666
  ```
- Check if your module is present in the introspection
- `curl -s http://127.0.0.1:6666/modules`
- If yes, test it directly with MISP or via curl
# Configuration at the top
moduleconfig = ['username', 'password']

# Code block in the handler
if request.get('config'):
    if (request['config'].get('username') is None) or (request['config'].get('password') is None):
        miserrors['error'] = 'CIRCL Passive SSL authentication is missing'
        return miserrors

    x = PyPSSL(basic_auth=(request['config']['username'], request['config']['password']))
Default Expansion Module Set

- asn history
- CIRCL Passive DNS
- CIRCL Passive SSL
- Country code lookup
- CVE information expansion
- DNS resolver
- DomainTools
- eupi (checking url in phishing database)
- IntelMQ (experimental)
- ipasn
- PassiveTotal - http://blog.passivetotal.org/misp-sharing-done-differently
- sourcecache
- Virustotal
- Whois
Similar to expansion modules
Input is a file upload or a text paste
Output is a list of parsed attributes to be editend and verified by the user
Some examples
  ▶ Cuckoo JSON import
  ▶ email import
  ▶ OCR module
  ▶ Open IoC import
Not the preferred way to export data from MISP
Input is currently only a single event
Output is a file in the export format served back to the user
Will be moved / merged with MISP built-in export modules
- Allows export of event / attribute collections
Backward compatible - an additional field to extend the format

```python
misp_attributes = {'input': [...], 'output': [...], 'format': 'misp_standard'}
```

Takes a standard MISP attribute as input

Returns MISP format

- Attributes
- Objects (with their references)
- Tags

```python
results = {'Attribute': [...], 'Object': [...], 'Tag': [...]}
```

First modules supporting this new export format

- urlhaus expansion module
- Joe Sandbox import & query module
### New Expansion & Import Modules View (MISP 2.4.110)

#### Enrichment Results

Below you can see the attributes and objects that are to be created from the results of the enrichment modules.

<table>
<thead>
<tr>
<th>Category</th>
<th>Type</th>
<th>Value</th>
<th>UUID</th>
<th>Tags</th>
<th>IDS</th>
<th>Disable Correlation</th>
<th>Comment</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payload delivery</td>
<td>sha256</td>
<td>d313ad9311b80be1d54c881ba2f3f3b02e6e17730170badf785718a4752zed</td>
<td>50378b09-80cd-4bd4-aadd-eb5e6432fb05b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Inherit event</td>
</tr>
<tr>
<td>Network activity</td>
<td>url</td>
<td><a href="http://automotive/dreamteam.com/vexe">http://automotive/dreamteam.com/vexe</a></td>
<td>e8765f00-cb67-4350-9620-20c3d9456d5c0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Inherit event</td>
</tr>
<tr>
<td>Network activity</td>
<td>url</td>
<td><a href="http://shopping/poop.fox.fox.com/vexe">http://shopping/poop.fox.fox.com/vexe</a></td>
<td>a3f85a11-4e6c-4d30-8a49-3e999650420bc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Inherit event</td>
</tr>
<tr>
<td>Network activity</td>
<td>url</td>
<td><a href="http://depoop.com/pooperfranchise.com/vexe">http://depoop.com/pooperfranchise.com/vexe</a></td>
<td>3d77be88-7c98-4195-ae52-7ad6a3d91ebc0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Inherit event</td>
</tr>
<tr>
<td>Network activity</td>
<td>url</td>
<td><a href="http://cherryinicpoopers.com/m/docs/vexe">http://cherryinicpoopers.com/m/docs/vexe</a></td>
<td>b804cb74-4af2-4e67-8db4-4ad988781411a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Inherit event</td>
</tr>
<tr>
<td>Network activity</td>
<td>url</td>
<td><a href="http://docs/poop.net/vexe">http://docs/poop.net/vexe</a></td>
<td>65067208-8288-4699-8c15-5331502266d4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Inherit event</td>
</tr>
<tr>
<td>Network activity</td>
<td>url</td>
<td><a href="http://alcohol/poop.mobi/vexe">http://alcohol/poop.mobi/vexe</a></td>
<td>34a7de0e-b736-47d0-94c1-d38307bbdb0ae</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Inherit event</td>
</tr>
<tr>
<td>Network activity</td>
<td>url</td>
<td><a href="http://alcohol/poop.info/vexe">http://alcohol/poop.info/vexe</a></td>
<td>058a135b-4f63-472b-aca8-0d22b4a6edab3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Inherit event</td>
</tr>
<tr>
<td>Network activity</td>
<td>url</td>
<td><a href="http://alcohol/poop.biz/vexe">http://alcohol/poop.biz/vexe</a></td>
<td>90cb3d6d-c775-4415-b544-5a20653ed47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Inherit event</td>
</tr>
</tbody>
</table>

**Event ID:** 1229

**Event UUID:** 5cc3042e-8bbd-4837-9564-47ace96451a

**Event creator org:** ORGNAME

**Event Info:** urhaus test

**Resolved Attributes:** 14 (2 Objects)
Future of the modules system

- Enrichment on full events
- Move the modules to background processes with a messaging system
- Have a way to skip the results preview
  - Preview can be very heavy
  - Difficulty is dealing with uncertain results (without the user having final say)
We welcome new modules and pull requests.

MISP modules can be designed as standalone application.