EXTENDING MISP WITH PYTHON MODULES

MISP - Threat Sharing

CIRCL / Team MISP Project

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

CIISI-IE
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
GOALS FOR THE MODULE SYSTEM

- 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 functionality 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.
Finding available MISP modules

```
curl -s http://127.0.0.1:6666/modules
```

```
{
  "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"
    ]
  }
}
```
### 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 enrichment services.</td>
</tr>
<tr>
<td>Recommended</td>
<td>Plugin.Enrichment_services_port</td>
<td>6666</td>
<td>The port used to access the enrichment services.</td>
</tr>
<tr>
<td>Recommended</td>
<td>Plugin.Enrichment_cve_enabled</td>
<td>false</td>
<td>Enable or disable the cve module.</td>
</tr>
<tr>
<td>Recommended</td>
<td>Plugin.Enrichment_dns_enabled</td>
<td>true</td>
<td>Enable or disable the dns module.</td>
</tr>
<tr>
<td>Recommended</td>
<td>Plugin.Enrichment_sourcecache_enabled</td>
<td>false</td>
<td>Enable or disable the sourcecache module.</td>
</tr>
<tr>
<td>Recommended</td>
<td>Plugin.Enrichment_sourcecache_archivepath</td>
<td></td>
<td>Set this required module specified.</td>
</tr>
<tr>
<td>Recommended</td>
<td>Plugin.Enrichment_passivetotal_enabled</td>
<td>true</td>
<td>Enable or disable the passivetotal module.</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 specified.</td>
</tr>
<tr>
<td>Recommended</td>
<td>Plugin.Enrichment_passivetotal_password</td>
<td></td>
<td>Set this required module specified.</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:

<table>
<thead>
<tr>
<th>Value</th>
<th>Category</th>
<th>Type</th>
<th>IDS</th>
<th>Comment</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>23.100.122.175</td>
<td><strong>Network activity</strong></td>
<td><strong>ip-src</strong></td>
<td></td>
<td>- imported via the freetext import</td>
<td><strong>X</strong></td>
</tr>
</tbody>
</table>

**Submit**

- **ip-src** ➔ **ip-dst**

**Update all comment fields**

**Change all**
MISP MODULES - MAIN TYPES OF MODULES

- 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

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

and the response of the dns module:

```json
{
  "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:
        misperrors['error'] = 'NXDOMAIN'
        return misperrors
    except dns.exception.Timeout:
        misperrors['error'] = 'Timeout'
        return misperrors
    except:
        misperrors['error'] = 'DNS resolving error'
    return misperrors

    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`

```bash
[ adulau:-/git/misp-modules/bin ]$ 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.PyPSSL(basic_auth=(request['config']['username'], request['config']['password']))
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 edited 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

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

Takes a standard MISP attribute as input

Returns MISP format

- Attributes
- Objects (with their references)
- Tags

```
  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 module.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>UUID</th>
<th>Tags</th>
<th>Disable Correlation</th>
<th>Comment</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event ID</td>
<td>1229</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event UUD</td>
<td>5cc3042c-b8d4-4837-9564-47ace956451a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event creator org</td>
<td>ORGNAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Info</td>
<td>urhaus test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iResolved Attributes</td>
<td>14 (2 Objects)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>Type</td>
<td>Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Payload delivery</td>
<td>sha256: sha256</td>
<td>5025b0f8-f0c6-4d04-a4d6-6b5e969e0f25b0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network activity</td>
<td>url</td>
<td><a href="http://automotive.screedteam.com/">http://automotive.screedteam.com/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network activity</td>
<td>url</td>
<td><a href="http://shopazlo.popsnop.co.in/">http://shopazlo.popsnop.co.in/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network activity</td>
<td>url</td>
<td><a href="http://pooper.scoopperfranchise.com/">http://pooper.scoopperfranchise.com/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network activity</td>
<td>url</td>
<td><a href="http://cherrynii.popapop.cooper.com/">http://cherrynii.popapop.cooper.com/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network activity</td>
<td>url</td>
<td><a href="http://logs.propool.net/">http://logs.propool.net/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network activity</td>
<td>url</td>
<td><a href="http://logs.propool.mob/V/">http://logs.propool.mob/V/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network activity</td>
<td>url</td>
<td><a href="http://logs.propool.info/V/">http://logs.propool.info/V/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network activity</td>
<td>url</td>
<td><a href="http://logs.propool.buzz/">http://logs.propool.buzz/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ID: 12700
Name: test
References: 1

---

MISP /two.osf/four.osf/one.osf/one.osf/zero.osf
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)
- https://github.com/MISP/misp-modules
- https://github.com/MISP/
- We welcome new modules and pull requests.
- MISP modules can be designed as standalone application.