Extending MISP with Python modules

MISP - Malware Information Sharing Platform & Threat Sharing

CIRCL
Computer Incident
Response Center
Luxembourg

Alexandre Dulaunoy
Andras Iklody

TLP:WHITE

June 16, 2016

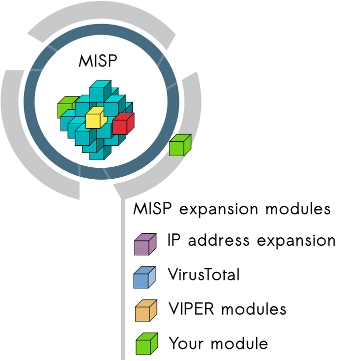
Why we want to go more modular...

- 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**

MISP modules - extending MISP with Python scripts



MISP

MISP expansion modules

- 🟪 IP address expansion
- 🟦 VirusTotal
- 🟧 VIPER modules
- 🟩 Your module

- 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 modules - installation

- MISP modules can be run on the same system or on a remote server.
- Python 3 is required to run MISP modules.
  - git clone git@github.com:MISP/misp-modules.git
  - cd misp-modules
  - pip3 install -r REQUIREMENTS
  - cd bin
  - python3 misp-modules.py

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"
  ]
}
```

Querying a module

- curl -s http://127.0.0.1:6666/query -H ”Content-Type: application/json” –data @body.json -X POST

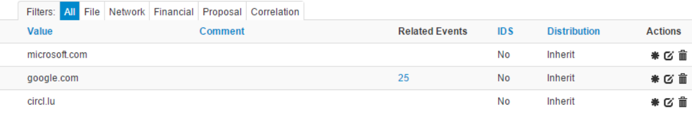
body.json

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

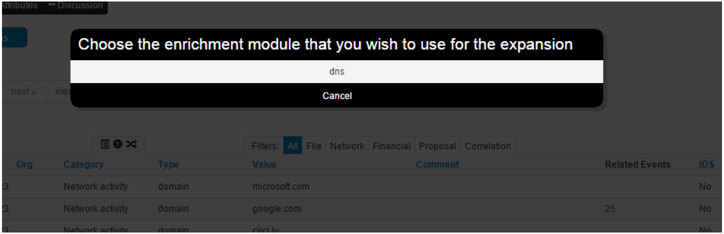
- and the response of the dns module:

```
{"results": [{"values": ["149.13.33.14"],
 "types": ["ip-src", "ip-dst"]}]}
```

MISP modules - How it’s integrated in the UI?

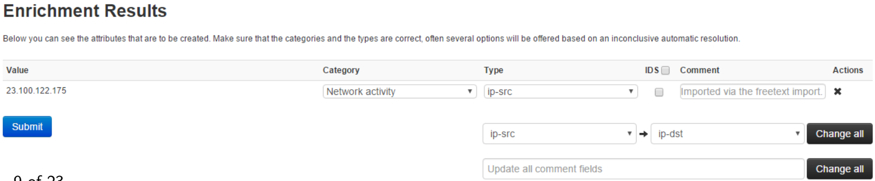


Value	Comment	Related Events	IDS	Distribution	Actions
microsoft.com			No	Inherit	
google.com		25	No	Inherit	
circl.lu			No	Inherit	



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

Org	Category	Type	Value	Comment	Related Events	IDS
3	Network activity	domain	microsoft.com			No
3	Network activity	domain	google.com		25	No
3	Network activity	domain	circl.lu			No



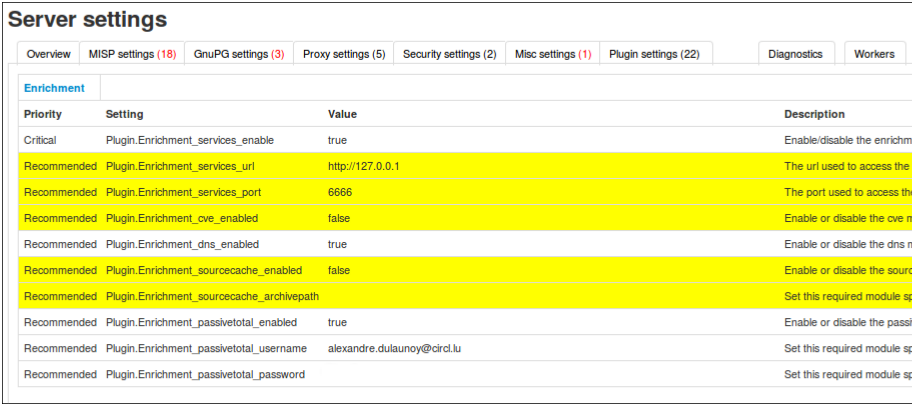
Enrichment Results

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.

Value	Category	Type	IDS □	Comment	Actions
23.100.122.175	Network activity ▾	ip-src ▾	□	Imported via the freetext import.	✖

Submit
ip-src ▾ → ip-dst ▾ Change all
Update all comment fields Change all

MISP modules - configuration in the UI



Server settings

Priority	Setting	Value	Description
Critical	Plugin.Enrichment_services_enable	true	Enable/disable the enrichm
Recommended	Plugin.Enrichment_services_url	http://127.0.0.1	The url used to access the
Recommended	Plugin.Enrichment_services_port	6666	The port used to access th
Recommended	Plugin.Enrichment_cve_enabled	false	Enable or disable the cve m
Recommended	Plugin.Enrichment_dns_enabled	true	Enable or disable the dns m
Recommended	Plugin.Enrichment_sourcecache_enabled	false	Enable or disable the sourc
Recommended	Plugin.Enrichment_sourcecache_archivepath		Set this required module sp
Recommended	Plugin.Enrichment_passivetotal_enabled	true	Enable or disable the passi
Recommended	Plugin.Enrichment_passivetotal_username	alexandre.dulaunoy@circl.lu	Set this required module sp
Recommended	Plugin.Enrichment_passivetotal_password		Set this required module sp

Creating your module (Skeleton)

```python
import json
import dns.resolver

misperrors = {'error' : 'Error'}
mispattributes = {'input': [], 'output': []}
moduleinfo = {'version': '', 'author': '',
              'description': '', 'module-type': []}

def handler(q=False):
    if q is False:
        return False
    request = json.loads(q)
    r = {'results': [{'types': [], 'values':[]}]}
    return r
def introspection():
    return mispattributes
def version():
    return moduleinfo
```

Creating your module (metadata 1)

```
misperrors = {'error' : 'Error'}
mispattributes = {'input': ['hostname', 'domain'], 'output': ['ip-src', 'ip-dst']}
moduleinfo = {'version': '', 'author': '',
              'description': '', 'module-type': []}
```

Creating your module (metadata 2)

```
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']}
```

Creating your module (handler 1)

```
def handler(q=False):
    if q is False:
        return False
    request = json.loads(q)
    # MAGIC
    # MORE MAGIC
    r = {'results': [
        {'types': output_types, 'values':values},
        {'types': output_types2, 'values':values2}
    ]}
    return r
```

Creating your module (handler 2)

```
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
```

Creating your module - finished DNS module

```
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

```
[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

Code samples (Configuration)

```
# 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):
            misperrors['error'] = 'CIRCL␣Passive␣SSL␣authentication␣is␣missing'
            return misperrors
_
    x = pypssl.PyPSSL(basic_auth=(request['config']['username'], request['config']['password']))
```

Default module set

- asn history
- CIRCL PassiveDNS
- CIRCL PassiveSSL
- CVE
- DNS
- eupi
- IntelMQ (experimental)
- ipasn
- PassiveTotal - http://blog.passivetotal.org/misp-sharing-done-differently
- sourcecache

Upcoming additions to the module system - Import modules

- Similar to enrichment 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 ideas for modules that we are looking into
  - Reimplementing some of the current imports
  - STIX 2.0 Import
  - OpenIOC Import
  - Connection to various sandboxes

Upcoming additions to the module system - Export modules

- Input initially will be an event
- Dynamic settings
- Later on to be expanded to event collections / attribute collections
- Output is a file in the export format served back to the user
- Some ideas for modules that we are looking into
  - STIX 2.0 Export
  - Bro export
  - ArcSight output

Upcoming additions to the module system - General

- Expose the modules to the APIs
- Move the modules to background processes with a messaging system
- Difficulty is dealing with uncertain results on import

Q&A

- 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.