

Information Sharing and Taxonomies

Practical Classification of Threat Indicators using MISP



CIRCL
Computer Incident
Response Center
Luxembourg

Alexandre Dulaunoy
Andras Iklody
Raphaël Vinot
TLP:WHITE

<http://www.misp-project.org/>
Twitter: @MISPProject

MISP Training Luxembourg
20170207

From Tagging to Flexible Taxonomies

OSINT - Cyberthreats BlackEnergy2

Event ID	2910
Uuid	568e7167-4e00-4654-b5f8-4b23950d210f
Org	CIRCL
Owner org	CIRCL
Contributors	
Email	alexandre.dulaunoy@circl.lu
Tags	ttp:white Type:OSINT +
Date	2016-01-07
Threat Level	Medium

- Tagging is a simple way to attach a classification to an event or an attribute.
- In the early version of MISP, tagging was local to an instance.
- **Classification must be globally used to be efficient.**
- After evaluating different solutions of classification, we build a new scheme using the concept of machine tags.

Machine Tags

- Triple tag or machine tag was introduced in 2004 to extend geotagging on images.

admiralty-scale:source-reliability="c"

namespace predicate value

- A machine tag is just a tag expressed in way that allows systems to parse and interpret it.
- Still have a human-readable version:
 - admiralty-scale:Source Reliability="Fairly reliable"

MISP Taxonomies

- Taxonomies are implemented in a simple JSON format.
- Anyone can create their own taxonomy or reuse an existing one.
- The taxonomies are in an independent git repository¹.
- These can be freely reused and integrated in other threat intel tools.
- Taxonomies are licensed under CC0 (public domain) except if the taxonomy author decided to use another license.

¹<https://www.github.com/MISP/misp-taxonomies/>

Existing Taxonomies

- NATO - **Admiralty Scale**
- CIRCL Taxonomy - **Schemes of Classification in Incident Response and Detection**
- eCSIRT and IntelMQ incident classification
- EUCI **EU classified information marking**
- Information Security Marking Metadata from DNI (Director of National Intelligence - US)
- NATO Classification Marking
- OSINT **Open Source Intelligence - Classification**
- TLP - **Traffic Light Protocol**
- Vocabulary for Event Recording and Incident Sharing - **VERIS**
- and many more like ENISA, Europol, or the draft FIRST SIG Information Exchange Policy.

Want to write your own taxonomy? 1/2

```
1 {
2   "namespace": "admiralty-scale",
3   "description": "The Admiralty Scale (also called the NATO
4     System) is used to rank the reliability of a source and
5     the credibility of an information.",
6   "version": 1,
7   "predicates": [
8     {
9       "value": "source-reliability",
10      "expanded": "Source Reliability"
11    },
12    {
13      "value": "information-credibility",
14      "expanded": "Information Credibility"
15    }
16  ],
17  ...
```





















Want to write your own taxonomy? 2/2

```
1 {  
2   "values": [  
3     {  
4       "predicate": "source-reliability",  
5       "entry": [  
6         {  
7           "value": "a",  
8           "expanded": "Completely reliable"  
9         }  
10    ],  
11  ]  
12 }
```

- Publishing your taxonomy is as easy as a simple git pull request on [misp-taxonomies](https://github.com/MISP/misp-taxonomies)².

²<https://github.com/MISP/misp-taxonomies>

How are taxonomies integrated in MISP?

10	✘	TO:HIDE		2	 
9	✘	TODO		8	 
11	✘	TODO:VT-ENRICHMENT		9	 
1	✔	Type:OSINT		932	 
18	✔	admiralty-scale:information-credibility="1"	admiralty-scale	0	 
19	✔	admiralty-scale:information-credibility="2"	admiralty-scale	1	 
20	✔	admiralty-scale:information-credibility="3"	admiralty-scale	3	 
21	✔	admiralty-scale:information-credibility="4"	admiralty-scale	0	 
22	✔	admiralty-scale:information-credibility="5"	admiralty-scale	1	 
23	✔	admiralty-scale:information-credibility="6"	admiralty-scale	2	 

- MISP administrator can just import (or even cherry pick) the namespace or predicates they want to use as tag.
- Tags can be exported to other instances.
- Tags are also accessible via the MISP REST API.

Filtering the distribution of events among MISP instances

- Applying rules for distribution based on tags:

Set push rules

Allowed Tags tip:white	Available Tags Type:OSINT tip:green tip:amber tip:ex:chr admiralty-scale:informatic	Blocked Tags circl:topic="finance"
Allowed Organisations CIRCL	Available Organisations ADMIN	Blocked Organisations

Other use cases using MISP taxonomies

- Tags can be used to set events or attributes for **further processing by external tools** (e.g. VirusTotal auto-expansion using Viper).
- Ensuring a classification manager **classifies the events before release** (e.g. release of information from air-gapped/classified networks).
- **Enriching IDS export** with tags to fit your NIDS deployment.
- Using **IntelMQ** and MISP together to process events (tags limited per organization introduced in MISP 2.4.49).

Future functionalities related to MISP taxonomies

- **Sighting** support (thanks to NCSC-NL) is integrated in MISP allowing to auto expire IOC based on user detection.
- Adjusting taxonomies (adding/removing tags) based on their score or visibility via sighting.
- Simple taxonomy editors to **help non-technical users** to create their taxonomies.
- **Filtering mechanisms** in MISP to rename or replace taxonomies/tags at pull and push synchronisation.
- More public taxonomies to be included.

PyTaxonomies

- **Python module** to handle the taxonomies
- **Offline** and online mode (fetch the newest taxonomies from GitHub)
- Simple **search** to make tagging easy
- Totally independant from MISP
- **No external dependencies** in offline mode
- Python3 only
- Can be used to create & **dump a new taxonomy**

PyTaxonomies

```
from pytaxonomies import Taxonomies
taxonomies = Taxonomies()
taxonomies.version
# => '20160725'
taxonomies.description
# => 'Manifest file of MISP taxonomies available.'
list(taxonomies.keys())
# => ['tlp', 'eu-critical-sectors', 'de-vs', 'osint', 'circl', 'veris',
#     'ecsirt', 'dhs-ciip-sectors', 'fr-classif', 'misp', 'admiralty-scale', ...]
taxonomies.get('enisa').description
# 'The present threat taxonomy is an initial version that has been developed on
# the basis of available ENISA material. This material has been used as an ENISA-internal
# structuring aid for information collection and threat consolidation purposes.
# It emerged in the time period 2012-2015.'
print(taxonomies.get('circl'))
# circl:incident-classification="vulnerability"
# circl:incident-classification="malware"
# circl:incident-classification="fastflux"
# circl:incident-classification="system-compromise"
# circl:incident-classification="sql-injection"
# ....
print(taxonomies.get('circl').machinetags_expanded())
# circl:incident-classification="Phishing"
# circl:incident-classification="Malware"
# circl:incident-classification="XSS"
# circl:incident-classification="Copyright issue"
# circl:incident-classification="Spam"
# circl:incident-classification="SQL Injection"
```

The dilemma of false-positive

- False-positive is a **common issue** in threat intelligence sharing.
- It's often a contextual issue:
 - false-positive might be different per community of users sharing information.
 - organization might have their **own view** on false-positive.
- Based on the success of the MISP taxonomy model, we build misp-warninglists.

MISP warning lists

- misp-warninglists are lists of well-known indicators that can be associated to potential false positives, errors or mistakes.
- Simple JSON files

```
1 {
2   "name": "List of known public DNS resolvers",
3   "version": 2,
4   "description": "Event contains one or more public DNS
5     resolvers as attribute with an IDS flag set",
6   "matching_attributes": [
7     "ip-src",
7     "ip-dst"
8   ],
9   "list": [
10    "8.8.8.8",
11    "8.8.4.4", ... ]
12 }
```

MISP warning lists

- The warning lists are integrated in MISP to display an info/warning box at the event and attribute level.
- Enforceable via the API where all attributes that have a hit on a warninglist will be excluded.
- This can be enabled at MISP instance level.
- Default warning lists can be enabled or disabled like **known public resolver**, **multicast IP addresses**, **hashes for empty values**, **rfc1918**, **TLDs** or **known google domains**.
- The warning lists can be expanded or added in JSON locally or via pull requests.
- Warning lists can be also used for **critical or core infrastructure warning**, **personally identifiable information...**

Q&A



- <https://github.com/MISP/MISP>
- <https://github.com/MISP/misp-taxonomies>
- <https://github.com/MISP/PyTaxonomies>
- <https://github.com/MISP/misp-warninglists>
- info@circl.lu (if you want to join one of the MISP community operated by CIRCL)
- PGP key fingerprint: CA57 2205 C002 4E06 BA70 BE89 EAAD
CFFC 22BD 4CD5