# Information Sharing and Taxonomies

Practical Classification of Threat Indicators using

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

MISP Training @ SUNET 20190206



#### FROM TAGGING TO FLEXIBLE TAXONOMIES



- 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.

- 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 CCo (public domain) except if the taxonomy author decided to use another license.

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

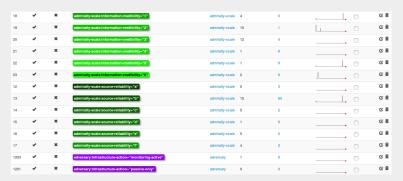
```
"namespace": "admiralty-scale",
     "description": "The Admiralty Scale (also called the NATO System
         ) is used to rank the reliability of a source and the
         credibility of an information.",
     "version": 1.
     "predicates": [
5
6
7
8
         "value": "source-reliability",
 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

# WANT TO WRITE YOUR OWN TAXONOMY? 2/2

■ Publishing your taxonomy is as easy as a simple git pull request on misp-taxonomies².

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

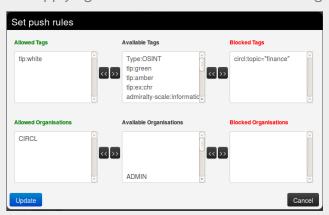
#### HOW ARE TAXONOMIES INTEGRATED IN MISP?



- 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:



#### 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 classies 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.kevs())
# => ['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="sal-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

#### 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 Coo2 4E06 BA70 BE89 EAAD CFFC 22BD 4CD5