Turning data into actionable intelligence
advanced features in MISP supporting your analysts and tools

CIRCL / Team MISP Project

Online Training (French)
The Computer Incident Response Center Luxembourg (CIRCL) is a government-driven initiative designed to provide a systematic response facility to computer security threats and incidents. CIRCL is the CERT for the private sector, communes and non-governmental entities in Luxembourg and is operated by securitymadein.lu g.i.e.
CIRCL is mandated by the Ministry of Economy and acting as the Luxembourg National CERT for private sector.

CIRCL leads the development of the Open Source MISP threat intelligence platform which is used by many military or intelligence communities, private companies, financial sector, National CERTs and LEAs globally.

CIRCL runs multiple large MISP communities performing active daily threat-intelligence sharing.
The aim of this presentation

- To give some insight into what sort of an evolution of our various communities’ have gone through as observed over the past 8 years
- Show the importance of **strong contextualisation**...
- ...and how that can be leveraged when trying to make our data **actionable**
There are many different types of users of an information sharing platform like MISP:

- **Malware reversers** willing to share indicators of analysis with respective colleagues.
- **Security analysts** searching, validating and using indicators in operational security.
- **Intelligence analysts** gathering information about specific adversary groups.
- **Law-enforcement** relying on indicators to support or bootstrap their DFIR cases.
- **Risk analysis teams** willing to know about the new threats, likelihood and occurrences.
- **Fraud analysts** willing to share financial indicators to detect financial frauds.
The initial scope of MISP

- **Extract information** during the analysis process
- Store and **correlate** these datapoints
- **Share** the data with partners
- Focus on technical indicators: IP, domain, hostname, hashes, filename, pattern in file/memory/traffic
- Generate protective signatures out of the data: snort, suricata, OpenIOC
Initial workflow

- Raw data → MISP Threat Sharing
- MISP Threat Sharing ↔ MISP Threat Sharing
- MISP Threat Sharing → Signatures
Why was it so simplistic?

- This was both a reflection of our maturity as a community
  - Capabilities for **extracting** information
  - Capabilities for **utilising** the information
  - Lack of **willingness** to share context
  - Lack of **co-operation** between teams doing technical analysis/monitoring and threat-intel

- The more growth we saw in maturity, the more we tried to match it with our data-model, often against pushback
There were separate factors that made our data-sets less and less useful for detection/defense in general

- **Growth of our communities**
- Distinguish between information of interest and raw data
- **False-positive** management
- TTPs and aggregate information may be prevalent compared to raw data (risk assessment)
- **Increased data volumes** leads to be able to prioritise
Our initial solution

- Allow users to tag any information created in MISP
- We wanted to be lax with what we accept in terms of data, but be strict on what we fed to our tools, with strong filter options
- We had some ideas on how to potentially move forward...
Our initial failures

- Try to capture different aspects of contextualisation into **normalised values** (threat level, source reliability, etc)
  - Didn’t scale with needs other than our own
  - Incorporating new types of contextualisation would mean **the modification of the software**
  - Getting communities with **established naming conventions** to use anything but their go-to vocabularies was a pipe-dream
  - Heated arguments over numeric conversions
We tried an alternate approach instead: Free tagging
  - Result was spectacularly painful, at least 7 different ways to spell tlp:amber
  - No canonisation for common terms lead to tagging ultimately becoming a highly flawed tool for filtering within a sharing community
How we ended up tackling the issue more successfully

- We ended up with a mixed approach, currently implemented by the MISP-taxonomy system
  - Taxonomies are **vocabularies** of known tags
  - Tags would be in a **triple tag format**
    
    ```
    namespace:predicate"value"
    ```
  - Create your own taxonomies, recipients should be able to use data you tag with them without knowing it at the first place
  - Avoid any coding, stick to **JSON**
- Massive success, approaching 100 taxonomies
- Organisations can solve their own issues without having to rely on us
Taxonomy tags often **non self-explanatory**

Example: universal understanding of tlp:green vs APT 28

For the latter, a single string was ill-suited

So we needed something new in addition to taxonomies - **Galaxies**

- Community driven **knowledge-base libraries used as tags**
- Including descriptions, links, synonyms, meta information, etc.
- Goal was to keep it **simple and make it reusable**
- Internally it works the exact same way as taxonomies (stick to **JSON**)

---

**Ransomware galaxy**

<table>
<thead>
<tr>
<th>Galaxy ID</th>
<th>373</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Ransomware</td>
</tr>
<tr>
<td>Namespace</td>
<td>misp</td>
</tr>
<tr>
<td>Uuid</td>
<td>344a2e-1480-4b6b-9aa8-9bb21341073</td>
</tr>
<tr>
<td>Description</td>
<td>Ransomware galaxy based on...</td>
</tr>
<tr>
<td>Version</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>777</td>
<td>Serleg</td>
</tr>
<tr>
<td>7ev3n</td>
<td>7ev3n-HONEST</td>
</tr>
</tbody>
</table>
Broadening the scope of what sort of context we are interested in

- **Who** can receive our data? **What** can they do with it?
- **Data accuracy, source reliability**
- **Why** is this data relevant to us?
- **Who** do we think is behind it, **what tools** were used?
- What sort of **motivations** are we dealing with? Who are the **targets**?
- How can we **block/detect/remediate** the attack?
- What sort of **impact** are we dealing with?
Parallel to the contextualisation efforts: False positive handling

- Low quality / false positive prone information being shared
- Lead to alert-fatigue
- Exclude organisation xy out of the community?
- False positives are often obvious - can be encoded
- Warninglist system\(^1\) aims to do that
- Lists of well-known indicators which are often false-positives like RFC1918 networks, ...

\(^1\)https://github.com/MISP/misp-warninglists
Atomic attributes were a great starting point, but lacking in many aspects.

**MISP objects** system
- Simple templating approach
- Use templating to build more complex structures
- Decouple it from the core, allow users to define their own structures
- MISP should understand the data without knowing the templates
- Massive caveat: **Building blocks have to be MISP attribute types**
- Allow relationships to be built between objects

---

²https://github.com/MISP/misp-objects
SUPPORTING SPECIFIC DATAMODEL
Continuous feedback loop

- Data ingested by MISP was in a sense frozen in time
- We had a creation data, but lacked a way to use the output of our detection
- Lead to the introduction of the *Sighting system*
- The community could sight indicators and convey the time of sighting
- Potentially powerful tool for IoC lifecycle management, clumsy query implementation default
Supporting specific datamodel

Events

<table>
<thead>
<tr>
<th>Sightings</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIRCL: 2 (2017-03-19 16:17:59)</td>
</tr>
</tbody>
</table>

| Tags |
| Date: 2016-02-24 |
| Threat Level: High |
| Analysis: Initial |
| Distribution: Connected communities |
| Sighting Details: freetext test |
| MISP: 2 |
| CIRCL: 2 |
Most obvious goal: Improve the way we query data

- Unified all export APIs
- Incorporate all contextualisation options into API filters
- Allow for an **on-demand** way of excluding potential false positives
- Allow users to easily **build their own** export modules feed their various tools
/attributes/restSearch
{
    "returnFormat": "netfilter",
    "enforceWarninglist": 1,
    "tags": {
        "NOT": [
            "tlp:white",
            "type:OSINT"
        ],
        "OR": [
            "misp-galaxy:threat-actor="Sofacy",
            "misp-galaxy:sector="Chemical"
        ]
    }
}
Make decisions on whom to share data with based on context
  ▶ MISP by default decides based on the information creator’s decision who data gets shared with
  ▶ Community hosts should be able to act as a safety net for sharing
    ■ Push filters - what can I push?
    ■ Pull filters - what am I interested in?
    ■ Local tags allow for information flow control
The emergence of ATT&CK and similar galaxies

- Standardising on high-level TTPs was a solution to a long list of issues
- Adoption was rapid, tools producing ATT&CK data, familiar interface for users
- A much better take on kill-chain phases in general
- Feeds into our filtering and situational awareness needs extremely well
- Gave rise to other, ATT&CK-like systems tackling other concerns
  - attck4fraud by Francesco Bigarella from ING
  - Election guidelines by NIS Cooperation Group

---

3https://www.misp-project.org/galaxy.html#_attck4fraud
4https://www.misp-project.org/galaxy.html#_election_guidelines
Example query to generate ATT&CK heatmaps

/events/restSearch
{
    "returnFormat": "attack",
    "tags": [
        "misp-galaxy:sector="Chemical"
    ],
    "timestamp": "365d"
}
A SAMPLE RESULT FOR THE ABOVE QUERY

![Screenshot of a table showing various cyber attack techniques and their mitigations](image-url)
Monitor trends outside of MISP (example: dashboard)
Decaying of indicators

We were still missing a way to use all of these systems in combination to decay indicators. Move the decision making from complex filter options to complex decay models. Decay models would take into account various taxonomies, sightings, the type of each indicator Sightings and Creation date. The first iteration of what we have in MISP now took:

- 2 years of research
- 3 published research papers
- A lot of prototyping
**Scoring Indicators: Our solution**

\[
\text{score}(\text{Attribute}) = \text{base\_score}(\text{Attribute, Model}) \cdot \text{decay}(\text{Model, time})
\]

Where,

- \(\text{score} \in [0, 100]\)
- \(\text{base\_score} \in [0, 100]\)
- decay is a function defined by model’s parameters controlling decay speed
- Attribute Contains Attribute’s values and metadata \((\text{Taxonomies, Galaxies, ...})\)
- Model Contains the Model’s configuration
IMPLEMENTATION IN MISP: Event/view

- Decay score toggle button
  - Shows Score for each *Models* associated to the *Attribute* type
Implementation in MISP: API result

/attributes/restSearch

"Attribute": [ 
{
    "category": "Network activity",
    "type": "ip-src",
    "to_ids": true,
    "timestamp": "1565703507",
    [...]
    "value": "8.8.8.8",
    "decay_score": [ 
    {
        "score": 54.475223849544456,
        "decayed": false,
        "DecayingModel": { 
            "id": "85",
            "name": "NIDS Simple Decaying Model"
        }
    }
}
### Decaying Models

<table>
<thead>
<tr>
<th>ID</th>
<th>Organization</th>
<th>Usable to everyone</th>
<th>Name</th>
<th>Description</th>
<th>Parameters {}</th>
<th>Formula</th>
<th># Assigned Types</th>
<th>Version</th>
<th>Enabled</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>1</td>
<td>✓</td>
<td>Phishing model</td>
<td>Simple model to rapidly decay phishing website.</td>
<td>{ &quot;lifetime&quot;: 3, &quot;decay_speed&quot;: 2.3, &quot;threshold&quot;: 30, &quot;default_base_score&quot;: 80, &quot;base_score_config&quot;: { &quot;estimative_language&quot;: 0.5, &quot;phishing&quot;: 0.5 } }</td>
<td>Polynomial</td>
<td>9</td>
<td>1</td>
<td>✓</td>
<td>▲ ▾ ▽ ❯</td>
</tr>
</tbody>
</table>

| 85 | 1            | ✗                  | NIDS Simple Decaying Model | Simple decaying model for Network Intrusion Detection System (NIDS). | { "lifetime": 120, "decay_speed": 2, "threshold": 30, "default_base_score": 80, "base_score_config": { "estimative_language": 0.25, "priority_level": 0.25, "retention": 0.25, "targeted_threat_index": 0.125, "false_positive": 0.125 } } | Polynomial | 13              | 1       | ✓       | ▲ ▾ ▽ ❯ |

Page 1 of 1, showing 2 records out of 2 total, starting on record 1, ending on 2

View, update, add, create, delete, enable, export, import
Create, modify, visualise, perform mapping
IMPLEMENTATION IN MISP: base_score TOOL

Adjust Taxonomies relative weights

Example

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Tags</th>
<th>Base score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute 1</td>
<td>admiralty-scale-information-credibility=&quot;5&quot;</td>
<td>0.0</td>
</tr>
<tr>
<td>Attribute 2</td>
<td>priority-level,baseline-minor,admiralty-scale-source-reliability=&quot;d&quot;,admiralty-scale-information-credibility=&quot;2&quot;</td>
<td>38.2</td>
</tr>
<tr>
<td>Attribute 3</td>
<td>priority-level,severe,admiralty-scale-source-reliability=&quot;2&quot;</td>
<td>84.6</td>
</tr>
</tbody>
</table>

Computation steps

<table>
<thead>
<tr>
<th>Tag</th>
<th>Computation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>priority-level,baseline-minor</td>
<td>0.46</td>
<td>11.62</td>
</tr>
<tr>
<td>admiralty-scale-source-reliability=&quot;d&quot;</td>
<td>0.27</td>
<td>0.80</td>
</tr>
</tbody>
</table>
Simulate Attributes with different Models
/attributes/restSearch
{
    "includeDecayScore": 1,
    "includeFullModel": 0,
    "excludeDecayed": 0,
    "decayingModel": [85],
    "modelOverrides": {
        "threshold": 30
    }
    "score": 30,
}
To sum it all up...

- Massive rise in **user capabilities**
- Growing need for truly **actionable threat intel**
- Lessons learned:
  - **Context is king** - Enables better decision making
  - **Intelligence and situational awareness** are natural by-products of context
  - Don’t lock users into your **workflows**, build tools that enable theirs
Contact us
▶ https://twitter.com/mokaddem_sami
▶ https://twitter.com/iglocskat

Contact CIRCL
▶ info@circl.lu
▶ https://twitter.com/circl_lu
▶ https://www.circl.lu/

Contact MISPPProject
▶ https://github.com/MISP
▶ https://gitter.im/MISP/MISP
▶ https://twitter.com/MISPPProject