

INTEROPERABILITY IN MISP

ENABLING A FLAWLESS STREAM OF INFORMATION

TEAM CIRCL
TLP:CLEAR

AUSCERT 2024



MISP
Threat Sharing

- The pivotal role of interoperability in threat intelligence sharing
- MISP Standard format: designed for interoperability
- Interoperability mechanisms
- Data feeding mechanisms

INTEROPERABILITY IN THREAT INTELLIGENCE SHARING

THE PIVOTAL ROLE OF INTEROPERABILITY IN THREAT INTELLIGENCE SHARING

- Ensuring a **seamless flow of information** between tools
 - ▶ Efficiency in information sharing
 - ▶ Enables faster dissemination of threat intelligence
- Enabling the scalability of the CTI pipeline with the integration of more tools
 - ▶ Flexibility in the choice of tools
 - ▶ More comprehensive view of threats
- Fostering **collaboration**
 - ▶ Encouraging the sharing of information
 - ▶ Can lead to faster response to threats

■ **Standardisation is key**

- ▶ Relying on **standard formats** is mandatory
- ▶ **Wide adoption** of these formats is highly encouraged
- ▶ **Conversion mechanisms** between formats are essential

■ Taking advantages of **automation tools**

- ▶ **Efficiency in detection and response** is highly dependent on automation
- ▶ **Automated conversion** between formats included in your CTI pipeline is crucial
- ▶ Providing automation mechanisms to all users is a vector for **more collaboration**

A GENERIC DATA FORMAT DESIGNED FOR INTEROPERABILITY

- **JSON** format
- Designed for **flexibility** and **extensibility**
- A combination of meta-models with **generic field names** to describe data structures
 - ▶ Flexible to allow the description of any kind of information in a structured manner
 - ▶ Adaptable to easily extend the format to new use-cases
- Ensuring **long term interoperability** with existing MISP software and other Threat Intelligence Platforms and tools

- Events as simple containers for embedded information
 - ▶ Can be an incident, a security analysis, a threat intelligence report, or anything else
 - ▶ No semantic meaning attached to the event itself
 - ▶ Meaning of an Event only **depends on the embedded information**
- Attributes as the granular pieces of information to describe IoCs
 - ▶ Made up of a **category - type - value** triplet
 - ▶ Category and type give meaning to the value
 - ▶ Difference between IoCs and observed data relies on a flag

- **Simple containers** grouping MISP Attributes to describe more complex data points
 - ▶ JSON format with generic meta information, such as the name and meta-category
 - ▶ The meaning of each Attribute within the object is defined by the object relation
- A generic templating system
 - ▶ Commonly used templates are provided by default
 - ▶ Easily **extensible** to new use-cases
 - ▶ Users can create **their own templates**
- Include a vocabulary to describe the various **inter object and object to attribute relationships**

- Taxonomies are ensuring the **consistency** of the tags used in MISP
 - ▶ Providing a **global classification** of data
 - ▶ **Reused by other tools** interacting with MISP
- MISP Galaxies provide a way to attach **more complex structures** to MISP data
 - ▶ They basically are tags with meta information
 - ▶ Describing known threat actors, malware, techniques or other collections of **contextual information**
 - ▶ MISP uses the tag name derived from the Galaxy Cluster
 - ▶ Support for **custom** Galaxy Clusters

THE SUPPORT OF FOCUSED SPECIFIC FORMATS

SUPPORTING SEVERAL PATTERNING LANGUAGES & SIGNATURE FORMATS

- Provide information on how data has been detected/extracted in addition to the actual data
- Including:
 - ▶ Yara & Sigma signatures
 - ▶ Snort / Suricata & Zeek (previously Bro) rules
 - ▶ STIX patterns
- Each of these formats is a **specific attribute type** in MISP
- Given rules, patterns and signatures can be extracted from MISP and **used to feed the respective tools**

SEVERAL AUTOMATION TOOLS TO SUPPORT INTEROPERABILITY

■ Export **data collections** from MISP

- ▶ Enabled for several data structures - Events, Attributes, Galaxies, etc.
- ▶ Default format is **MISP standard - JSON**
- ▶ Supports a wide range of other formats, including CSV, XML, Yara, etc.
- ▶ **Advanced filtering capabilities**
- ▶ RESTfull API queries can be **automated** with *curl* commands or *Python* scripts using **PyMISP**

■ Import data into MISP Events

- ▶ **Lossless** MISP JSON Events ingestion
- ▶ **PyMISP** can parse different formats too and convert data into MISP format

AN ADVANCED STIX CONVERSION FEATURE

- Works as a **built-in module**
 - ▶ Convert any data collection to STIX
 - ▶ Import STIX files into MISP
- Supporting all STIX versions
 - ▶ STIX 1.x - XML
 - ▶ STIX 2.x - JSON
- Continuous development on STIX 2.x to **improve the conversion capacities** following evolutions on the STIX standards as well as the extensions of the MISP standard format
- Filling the mapping gaps over time to **improve interoperability** between MISP and other tools supporting STIX, such as TAXII, or STIX feeds producers

- **Simple Python scripts** to automate the **import/export** of data
 - ▶ Extending the range of supported formats
 - ▶ Allows anyone to build their own module to either:
 - Populate MISP Events with data from external sources/formats
 - Extract and convert data from MISP Events
- **Enrichment modules**
 - ▶ Use-case examples:
 - **enrich** data with additional context
 - **cross-reference** data with external sources
 - **validate** data
 - ▶ Can be triggered automatically by **Workflows**

MISP WORKFLOWS

■ Needs that Workflows can address:

- ▶ Prevent default MISP behaviors
- ▶ Trigger specific actions to run callbacks



- ZeroMQ channels
 - ▶ N-to-N Asynchronous message-processing tasks
 - ▶ Publisher(MISP) and consumer (scripts)
- **Streaming data as it is created in MISP**
- Advantage is the subscriber can **automatically use the published data**
- Be careful though with data being **republished**
- Also, there is **no access control** on the data that is streamed

DATA FEEDING MECHANISMS

- **Synchronisation is the default communication mechanism between MISP instances**
 - ▶ Exchange of MISP standard format
 - ▶ **Bidirectional** communication
 - ▶ **Filtering** capabilities
- Multiple data structures can be synchronised
 - ▶ **Events are synchronised by default** with their **Attributes & Objects**
 - ▶ Synchronisation of Galaxy Clusters, Analyst Data & Sightings can be enabled/disabled

■ **2-Step** process when Pulling Events

- ▶ Caching of the data
 - Lookup of the Events in the remote instance
 - Correlations with the Attributes in my instance
- ▶ Fetching data
 - Pulling the Events with their content on my instance

■ Automated pushing mechanism

- ▶ **Published Events** and their content are pushed to the remote instance(s)
- ▶ Users can manually push Events

- MISP Feeds provide a way to:
 - ▶ **Exchange information via any transport method** (HTTP, TLP, USB key, etc.)
 - ▶ Preview events along with their attributes, objects
 - ▶ Select and import events
 - ▶ **Correlate attributes using caching**
- Feeds work without the need of MISP synchronisation
- **Feeds can be produced without the need of a MISP instance**

■ References on the presented topics

- ▶ MISP Standards: <https://www.misp-standard.org/standards/>
- ▶ MISP Concepts Cheat sheet: <https://www.misp-project.org/misp-training/cheatsheet.pdf>
- ▶ MISP Feeds: <https://www.misp-project.org/misp-training/a.3-misp-feed.pdf>

■ More details on MISP

- ▶ Contact: info@circl.lu
- ▶ <https://www.misp-project.org>
- ▶ <https://github.com/MISP>