

MISP: Introduction, Concepts and Guide

This eLearning module is a prerequisite or refreshing module to read before the actual training sessions. This helps to ensure that all participants are inline with the basic knowledge of MISP. In the training modules, the various elements mentioned in this introduction will be completed in details (e.101-104, e.205-e.206 and e.302-e.304).

Structure of this document

1. **MISP Introduction:** The what, why and how about MISP
2. **MISP Basics:** A concise introduction to MISP data model
3. **How-to:** A user guide with screenshots on how to use MISP to encode and share data

MISP Introduction

What is MISP

MISP is an open-source threat-intelligence and sharing platform meant to store, correlate, enrich, analyse and share information. It enables the various type of analysts to collaborate on investigations and incidents, perform intelligence as well as helping operators to automatically feed their protective tools.

Why is MISP relevant

Information sharing is becoming more essential than ever to oppose threats. MISP strive to be the enabler and interface for real cross-sectoral sharing and support the organisations facing hybrid threats. To achieve these goals, MISP uses a practical information sharing format expressed in JSON which is built from a practical use-cases. It is flexible and can be easily extended by users to model their own data-structure.

The MISP core format as well as the common set of vocabularies provided by the various libraries supported by the tool allows users from all around the world to understand each others and rely on normalized data, making MISP a central place to collaborate.

MISP offers different alternatives to share analysis, case and report enabling users to review data produced by partners or third-parties and propose changes if need be. This happens in a decentralized way where analyst can evaluate correlation against other existing evidences and perform enrichment on the data.

These functionnalities provide the means to fulfill the ultimate goal of MISP: Bridging communities together. By fostering communication and sharing across multiple sectors, people are able to share and collaborate seemlessly making the connection between law enforcement with CSIRTs possible.

MISP philosophy

Sharing being the principal functionnality, it is essential that everyone is able to send and receive data. As such, everyone is considered to be a producer (also called contributor) and/or a consumer at the same time. There are strictly no obligation to contribute which in turns makes the system to have a low barrier of access for users to get acquainted to the system.

MISP Basics

A cheat-sheet describing the core concepts and data-models in MISP is available [here](#).

1.1 Data Layer

First and foremost, it's important to understand how MISP is organised. Similar to all applications, some predefined data structure exists and are used to represent and save the actual data on the disk. Such structure in MISP could be for example *Attributes* or *MISP Objects*.

MISP Attributes *Attributes* are individual block containing the very information to be used or to be shared. Thanks to their characteristic called **type**, *Attributes* can represent concept such as an IP address, a domain name or cryptographic hash. In addition to having a **type** and a **value**, they can express if they are Indicators of Compromise (IoC) or supporting data where for example, the former could be a hash of a malicious binary and the later could be Observed behaviour or links toward documentation. The differentiation between IoC and observable can be done by flipping the *Attribute*'s **to_ids** flag.

<input type="checkbox"/> 2021-11-25	Payload delivery	ip-src	118.217.182.3
<input type="checkbox"/> 2021-11-25	Payload delivery	url	https://evilprovider.com/this-is-not-malicious.exe

Figure 1: attributes

MISP Objects In most of the case, these individual blocks of information can be combined together into a more elaborated concept. When multiple *Attributes* are grouped, they form another entity that is called a *MISP Object*. For example, a *File Object* contains multiple *Attributes* such as the filename, its size, its name and so on.

By their very nature, *MISP Objects* organise and facilitate the reading of data in the application. But their efficiency can be improved even more when you add the capability to link them together with relationships to create directed graph

allowing to represent stories, processes or behaviours. In MISP, creating such connections is called “create an *Object Reference*”. Viewing these relationships as a connected graph can be done by looking at the widget called *Event Graph*.

The screenshot shows a list of objects and their references. At the top, there is a summary for the object "file" (ID 15) with a timestamp of 2021-12-09. It shows 1 reference and 1 reference. Below this, a table lists various payload delivery events and other objects, each with their type, attribute name, value, and a unique identifier.

Date	Type	Attribute	Value	Identifier
2021-12-09	Payload delivery	malware-sample:	malicious.exe	f1a3e62de12faecee82bf4599cc1fdcd
2021-12-09	Payload delivery	filename:	malicious.exe	
2021-12-09	Payload delivery	md5:	md5	f1a3e62de12faecee82bf4599cc1fdcd
2021-12-09	Payload delivery	sha1:	sha1	d836f2ee449b74913d1efc615eeb459b65e4f791
2021-12-09	Payload delivery	sha256:	sha256	d90401420908dbb4b3488a306467e8fffc57577ce9d5eee016578ff6a3ada1
2021-12-09	Other	size-in-bytes:	751328	

Figure 2: objects

MISP Events Now that we have the structures to encode information, we need another structure to be able to group them together in order to avoid dealing with a soup of *Attributes* and *MISP Objects*. *MISP Events* or commonly called *Events* are envelopes allowing to assemble *Attributes* and *Objects* contextually linked. Typically, *Events* are used to encode incidents, events or reports.

The screenshot shows the details of an event (ID 15). The left sidebar contains navigation links like View Event, View Connection Graph, and View Event History. The main area displays the event's ID, UUID, creator organization, owner organization, and creator user. It also shows tags, threat level, analysis, distribution, and info. A large red box highlights the "Published" field set to "No". Below this, the "Attributes" section lists 12 attributes, including first recorded change (2021-11-25), last change (2022-04-06 20:43), and a modification map. The bottom section shows sightings and a timeline table.

Date	Org	Category	Type	Value	Tags	Galaxies	Comment	Correlate	Related Events	Feed Hits	IDS	Distribution	Sightings	Activity	Actions
2022-03-04		Object name	File												
2021-11-25		Payload delivery	md5	079240ee4c0b2a22267feab7febe0	infected	malicious	malicious					infected			
2022-03-04		Payload delivery	filename	malicious.exe								infected			

Figure 3: event

Threat Intelligence Tools: Event Graph, Event Timeline and Event Reports

MISP Event Graph The MISP *Event Graph* feature is a widget accessible when viewing an *Event*. It allows analysts to visualise or create relationships between different entities in order to describe in a concise manner complex scenarios such as events performed in parallel or multiple-step attacks.

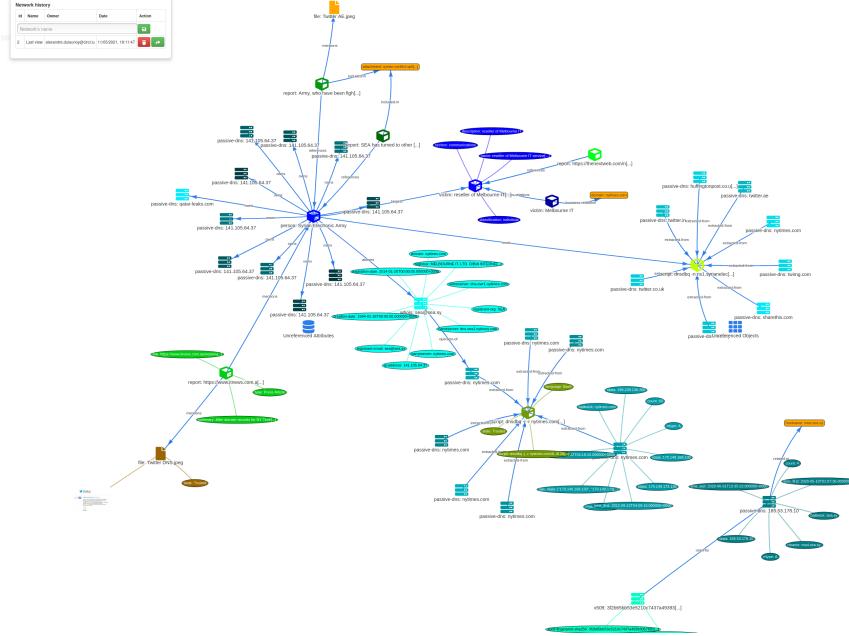


Figure 4: event-graph

MISP Event Timeline In some situations, temporality is crucial to understand the order of events, actions or processes. To help analysts visualise and adjust the time component of *Attributes* or *Objects*, a complete timeline viewer and editor is available allowing users to describe complex time-based information.

MISP Event Reports In addition to encode data into pre-formatted structure, MISP offers a tool to write report. Such reports are called *Events reports* and are contained in an *Event* where they use the markdown syntax to write formatted text. They also provide directives specific to MISP allowing writers to reference other entities contained in the *Event*. This extended syntax supports referencing *Attributes*, *Objects*, *Tags* and *Galaxy Clusters*.

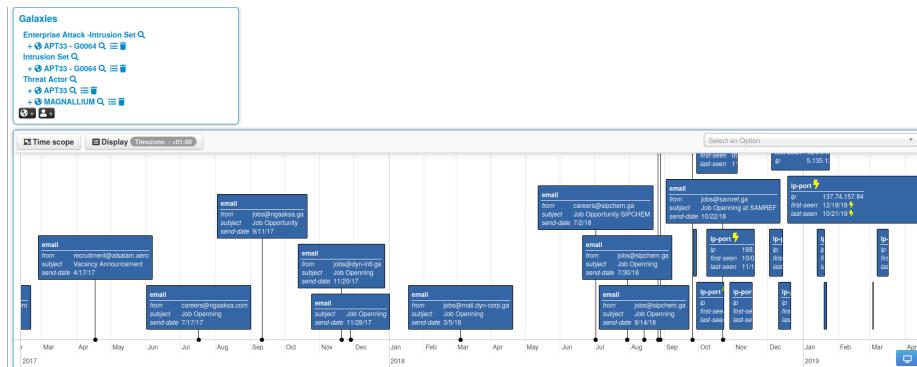


Figure 5: event-timeline

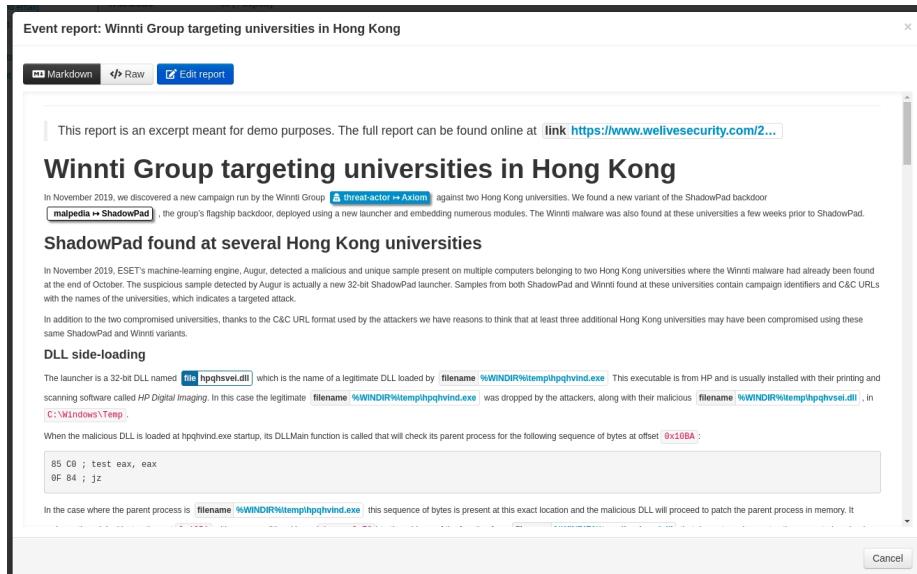


Figure 6: event-report

1.2 Context Layer

One of the most critical aspects often left aside is contextualisation. If done properly, it allows the reader to know more about where this data comes from, what it is about, how relevant it is for the user and finally, what can be done with it.

In MISP, contextualising data is as simple as attaching a label to the relevant entity. However, choosing the right labels is the difficult part. We can distinguish two types of labels: *Tags* and *Galaxy Clusters*.

Tags *Tags* are simple labels coming from a curated list of vocabulary (Also called *Taxonomy*). They are mainly used to classify data in order to ease data consumption and automation. For example, the following *Tags* can be used to quickly classify information: - **tlp**: Allow a favorable classification scheme for sharing sensitive information while keeping the control over its distribution at the same time. - **adversary**: An overview and description of the adversary infrastructure and allowed actions - **collaborative-intelligence**: Common language to support analysts to perform their analysis. The objective of this language is to advance collaborative analysis and to share earlier than later. - **estimative-language**: Estimative language to describe quality and credibility of underlying sources, data, and methodologies

Name	Expanded	Numerical Value	# Events	# Attributes	Tag	Enabled	Actions
estimative-language:confidence-in-analytic-judgment,"high"	Confidence in analytic judgment: High	95	15	4	estimative-language:confidence-in-analytic-judgment,"high"	✓	✖️✖️
estimative-language:confidence-in-analytic-judgment,"low"	Confidence in analytic judgment: Low	0	9	0	estimative-language:confidence-in-analytic-judgment,"low"	✓	✖️✖️
estimative-language:confidence-in-analytic-judgment,"moderate"	Confidence in analytic judgment: Moderate	55	26	4	estimative-language:confidence-in-analytic-judgment,"moderate"	✓	✖️✖️
estimative-language:likelihood-probability,"almost-certain"	Likelihood or probability: Almost certainty - nearly certain - 95-99%	95	21	8	estimative-language:likelihood-probability,"almost-certain"	✓	✖️✖️
estimative-language:likelihood-probability,"almost-no-chance"	Likelihood or probability: Almost no chance - remote - 01-05%	0	0	0	estimative-language:likelihood-probability,"almost-no-chance"	✓	✖️✖️
estimative-language:likelihood-probability,"likely"	Likelihood or probability: Likely - probable (probable) - 55-80%	55	4	5	estimative-language:likelihood-probability,"likely"	✓	✖️✖️
estimative-language:likelihood-probability,"roughly-even-chance"	Likelihood or probability: Roughly even chance - roughly even odds - 45-55%	45	4	2	estimative-language:likelihood-probability,"roughly-even-chance"	✓	✖️✖️
estimative-language:likelihood-probability,"unlikely"	Likelihood or probability: Unlikely - improbable (improbable) - 20-45%	20	0	1	estimative-language:likelihood-probability,"unlikely"	✓	✖️✖️
estimative-language:likelihood-probability,"very-likely"	Likelihood or probability: Very likely - highly probable - 80-95%	80	23	18	estimative-language:likelihood-probability,"very-likely"	✓	✖️✖️
estimative-language:likelihood-probability,"very-unlikely"	Likelihood or probability: Very unlikely - highly improbable - 05-20%	5	0	1	estimative-language:likelihood-probability,"very-unlikely"	✓	✖️✖️

Figure 7: taxonomy

Galaxy Clusters *Galaxy Clusters* are knowledge base items having descriptions, links, synonyms and any other meta-information. *Clusters* are regrouped into a higher-level structure called *Galaxy*. *Clusters* enable analysts to assign complex high-level contextual information to data-structures. Example of *Galaxy Clusters*:

- **threat-actor="Sofacy"** having information such as suspected-state-sponsor, victims, links-to-documentation, target-category and synonyms.
- **country="Luxembourg"** having information such as country-code, languages, TLD, Capital and so on.

MITRE's ATT&CK Another advantage that *Galaxy Clusters* have compared to simple labels is the fact that the list of *Clusters* belonging to the same *Galaxy* can be arranged as a matrix to have improved readability and aggregation. One of the biggest success of this kind of matrices is definitely the MITRE

Country :: luxembourg		
Name	luxembourg	
Parent Galaxy	Country	
Description	Luxembourg	
Version	1	
UUID	84668357-5a8c-4bdd-9f0f-6b50b24c5558	
Source	MISP Project	
Authors	geonames.org	
Distribution	All communities	
Creator Organisation	MISP	
Connector tag	misp-galaxy.country="luxembourg"	
Events	11 events	
« previous next »		
Tabular view JSON view		
Key ↓	Value	Actions
Capital	Luxembourg	
Continent	EU	
CurrencyCode	EUR	
CurrencyName	Euro	
ISO	LU	
ISO3	LUX	
Languages	lb,de-LU,fr-LU	
Population	497538	
tld	.lu	

Figure 8: cluster-country

ATT&CK framework. It describes tactics, techniques and procedures of adversaries. ATT&CK is very popular and its usage is highly recommended as it offers very precise classification and is globally understood and supported by other tools.

1.3 Anatomy of a complete Event

1.4 Distribution Levels

Distribution level is the term used in MISP to determine who can read which data and how it should be shared. The distribution can be set on entities such as *Event* or *Attributes*, where the most restrictive priority will always take priority.

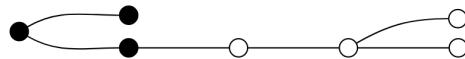
There are 5 distribution levels controlling who can see and how it should be shared:

- **Organisation only:** Only members of your organisation
- **This Community:** Organisations on one MISP instance
- **Connected Community:** Organisations on one MISP instance and those on MISP instances synchronising with this one. Upon receiving data, the distribution will be downgraded to *This community* to avoid further propagation

$n = 0$ $n = 1$ $n = 2$ $n = 3$ $n = 4$

Does not have the Event

Has the Event



Pre Attack - Attack Pattern		Enterprise Attack - Attack Patterns		Mobile Attack - Attack Pattern		Initial access		Exfiltration	
		Execution	Persistence	Privilege escalation	Defense evasion	Credential access	Discovery	Lateral movement	Collection
Spoofering Attachment	Scripting	Screensaver	Riley Sanction Permissions Weakness	Process Hollowing	SecurityId Memory	Password Discovery	AppleScript	Data from Information Repositories	Exfiltration Over Alternative Protocol
	Scripting via Service	Command-Line Interface	Login Item	AppCert DLLs	Code Signing	Input Capture	System Network Configuration Discovery	Data from Removable Media	Standard Application Layer Protocol
Trusted Relationship	User Execution	Trap	Application Shimming	Rootkit	Bash History	Process Discovery	Pass the Hash	Man in the Browser	Communication Through Removable Media
	Replication Through Removable Media	RegExs/Regasm	System Firmware	Scheduled Task	NTFS File Attributes	Exploitation for Credential Access	Network Share Discovery	Data Staged	Data Compressed
Exploit Public Facing Application	Trusted Developer Utilities	Registry Run Keys / Start Folder	Startup Items	Exploit for Defense Evasion	Private Keys	Peripheral Device Discovery	Exfiltration of Remote Services	Automated Exfiltration	Multi-Stage Channels
	Spoofering Link	Windows Management Instrumentation	LCI, LOADLIB, Addition	New Service	Force Brush	Account Discovery	Screen Capture	Scheduled Transfer	Remote Access Tools
Valid Accounts	Service Location	LSASS Driver	Auto Caching	Process Doppelgänging	Password Filter DLL	System Internation Discovery	Email Collection	Data Encrypted	Uncommonly Used Port
	Supply Chain Compromise	CMSIPT	Re:common	Process Injection	Disabling Security Tools	Two-Factor Authentication Interception	Windows Remote Management	Clipboard Data	MultiLayer Encryption
Drive-by-Compromise	Control Panel Items	Authentication Package	Bypass User Account Control	Timestamp	LLMNR/NBTNS Poisoning	System Network Connections Discovery	Windows Admin Shares	Video Capture	Exfiltration Over Physical Medium
	Hardware Additions	Dynamic Data Exchange	Component Firmware	Extra Window Memory Injection	Moshi Registry	Network Service Scanning	Remote Services	Audio Capture	Domain Fronting
Space after Firewall	Source	Windows Management Instrumentation Event Subscription	Seuid and Segid	Indicator Removal Tools	Credentials in Files	File and Directory Discovery	Taint Shared Content	Data from Network Shared Drive	Data Obfuscation
	Change Default File	Launch Daemon	Hidden Window	Keychain	Software Discovery	Application Deployment Software	Data from Local System	Connection Proxy	Commonly Used Port

Figure 9: cluster-country

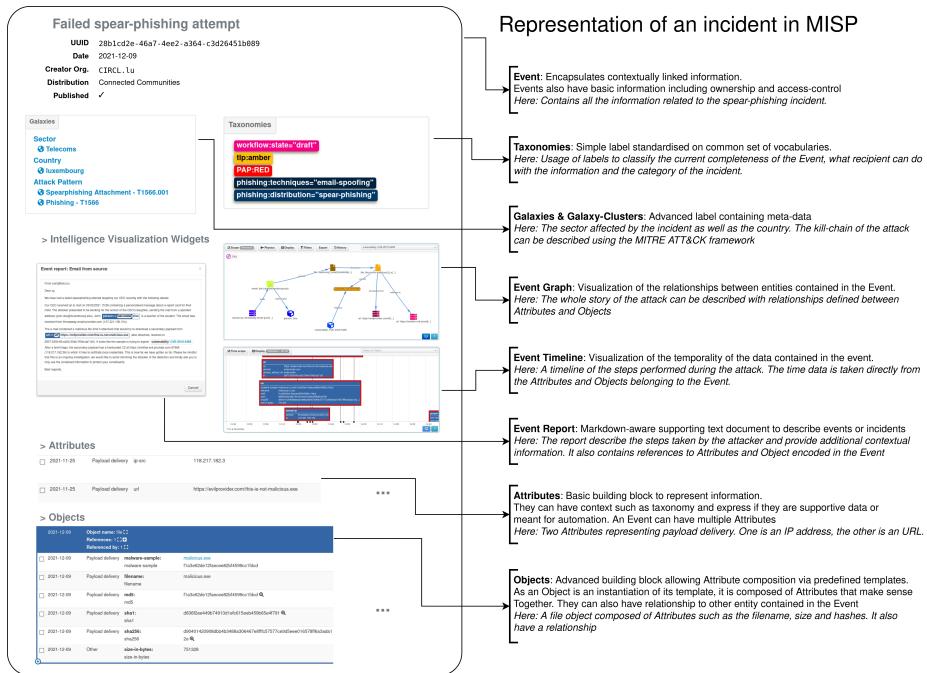
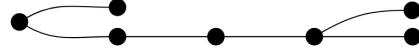


Figure 10: event-anatomy

- **All Community:** Anyone having access. Data will be freely propagated in the

$n = 0 \quad n = 1 \quad n = 2 \quad n = 3 \quad n = 4$

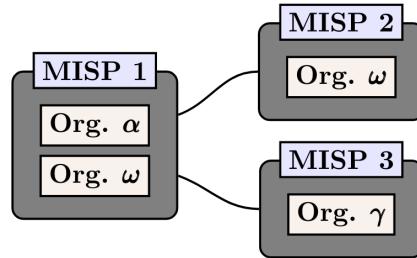


network of connected MISP instances

- **Sharing Groups:** Distribution list that exhaustively keeps track of which organisations can access the data and to which server it should be synchronised

Sharing Group configuration	
Organisations	Org. α Org. ω Org. γ
Instances*	MISP 1 MISP 2 MISP 3

*Or enable roaming mode instead



1.5 Synchronisation

In MISP, a synchronisation is the act of sharing data from one MISP to another. It can be done with two mechanisms, namely *push* and *pull*. The fact of an instance sending data to another is called *pushing*. If one instance retrieve data from another, it is called *pulling*.

The diagram below shows a one-way synchronisation link between two MISP instances. The Organisation α created a *sync_user* (denoted with a +) on MISP 2. A synchronisation link can be created on MISP 1 using the API Key and the organisation of the *sync_user*. At that point, MISP 1 can *pull* data from MISP 2 and can *push* data to MISP 2.

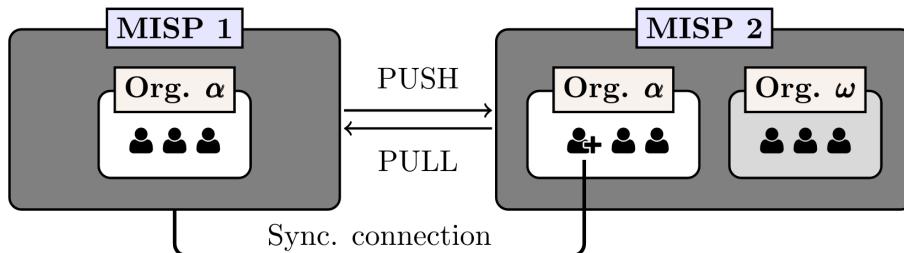


Figure 11: synchronisation

Once a synchronisation link exists *Events* can flow through that connection if and only if the distribution level of the *Event* allows it and if the *Event* is published.

1.6 Correlation

A *correlation* is a link between two *Attributes* that are created automatically. They allow interconnection between *Events* based on the correlation *Attribute*'s value. The system responsible to create these links is called the correlation engine and support not only strict string comparison but also more clever data type such as CIDR blocks and Fuzzy hashing like SSDEEP.

The correlation system is a tool meant for analysts to corroborate findings and gauge the trustiness of the data. It allows to confirm certain aspect of a report or to find new or unknown threats.

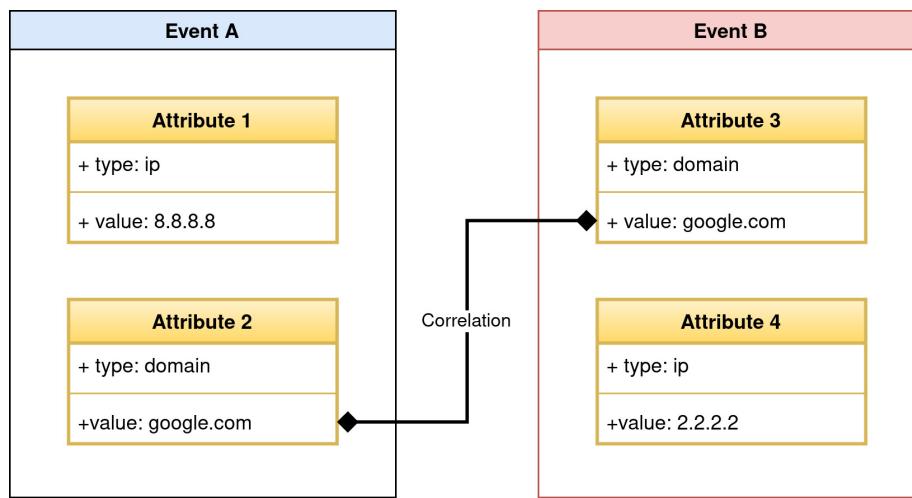


Figure 12: correlation

How-to

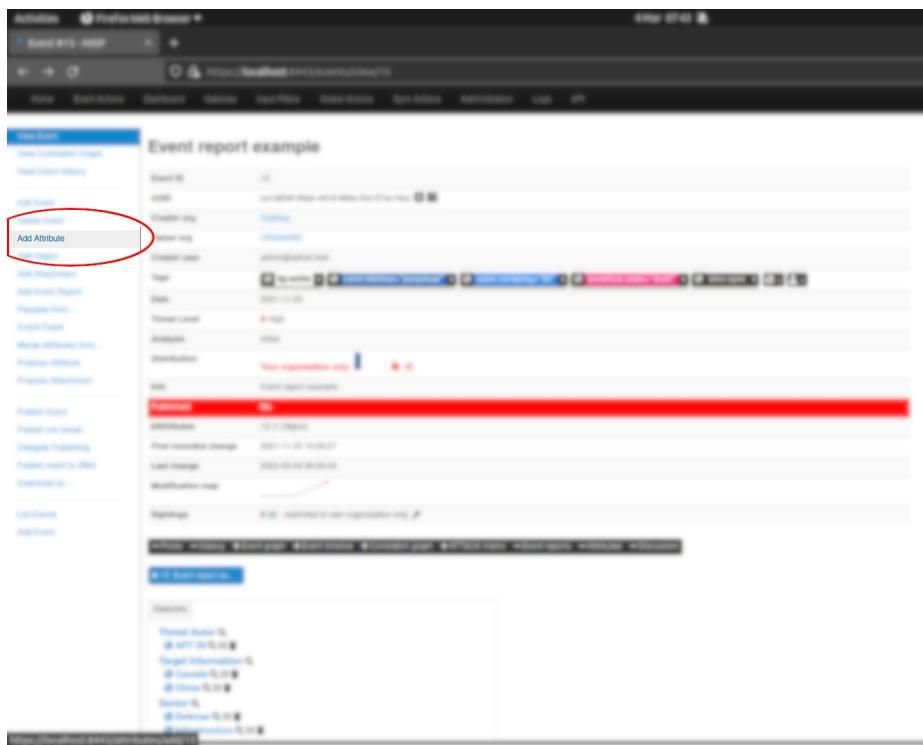
Create an Event

1. On the top-bar, click on **Event Action** then **Add Event**
2. Choose the correct distribution
3. Fill the **Event info** field with a concise summary of what this *Event* is about
4. Fill the remaining optional fields
5. Click on **Submit**

The screenshot displays the EventIDE web application interface. At the top, there's a navigation bar with links like Home, Open Actions, Dashboard, Devices, Input/Flow, Global Actions, Open Actions, Devices, Log, API. Below the navigation is a sidebar with a tree view of event categories: Log Overview, Add Attribute, Log Metadata, User Proposals, Events with proposals, View delegation responses, Events, and Communities. The main content area shows a detailed event timeline for an event named 'Training' on 2023-12-01. The timeline includes sections for Target Information, Response, and Detection. A legend indicates that green boxes represent 'This community only' and blue boxes represent 'Community-wide'. Below the timeline is a table listing various event details such as Date, Time, Type, and Community ID. At the bottom of the main content is a form titled 'Add Event' with fields for Date (2023-12-01), Distribution (set to 'This community only'), Thread Level (High), Analysis (Initial), and Event Info (containing 'Quick Event Description or Tracking Info'). There's also an 'Extend Event' section and a 'Submit' button.

Create an Attribute

1. When viewing an *Event*, click on **Add Attribute**
2. Fill the required **Category**, **Type** and **Value** field
3. Check **For Intrusion Detection System** checkbox if you consider this *Attribute* to be an indicator
4. Fill the remaining optional fields
5. Click on **Submit**



Add Attribute

Category i

Type i

Network activity

ip-src

Value

8.8.8.8

Contextual Comments

For Intrusion Detection System

Block source

Block destination

Last seen time ■

First seen time ■

Last seen time ■

First seen time ■

Last seen time ■

Submit

The screenshot shows a user interface for adding an attribute. At the top, there are dropdown menus for 'Category' (set to 'Network activity') and 'Type' (set to 'ip-src'). Below these is a text input field containing the value '8.8.8.8'. Underneath the value input is a section titled 'Contextual Comments' with a checked checkbox labeled 'For Intrusion Detection System'. There are also two timestamp inputs: 'Last seen time' and 'First seen time', each with a small calendar icon. At the bottom of the form is a prominent blue 'Submit' button.

Create an Object

1. When viewing an *Event*, click on **Add Object**
2. If you know the category of the *Object*, select it, otherwise pick **All Objects**
3. To add a “File” *Object*, search the entry in the dropdown or start typing **file** then select the entry

4. Fill out at least the requirements for this Object and additional other *Attributes*
5. Click on **Submit**
6. Review the *Object* you are about to create then it **Create new object**

Event report example

Object ID: 11
UUID: 2a1d9200-0000-4000-8000-000000000000

All Objects
Select an Option
file
cytomic-onion-file
File
network-profile
original-imported-file
regripper-software-hive-userprofile-winlogon

First seen date: 2023-01-01 00:00:00
Last seen date: 2023-01-01 00:00:00
Modification date: 2023-01-01 00:00:00
Pathage: 0% (estimated to run regeneration only)

Add File Object

Object template: File

Requirements
Required one of: filename, size-in-bytes, authentihash, ssdeep, md5, sha1, sha224, sha256, sha384, sha512, sha512/224, sha512/256, sha3-224, sha3-256, sha3-384, sha3-512, tlsh, telfhash, imphash, pattern-in-file, certificate, malware-sample, attachment, path, fullname

Meta category: File
Distribution: Inherit event
Comment:

First seen date: Last seen date:

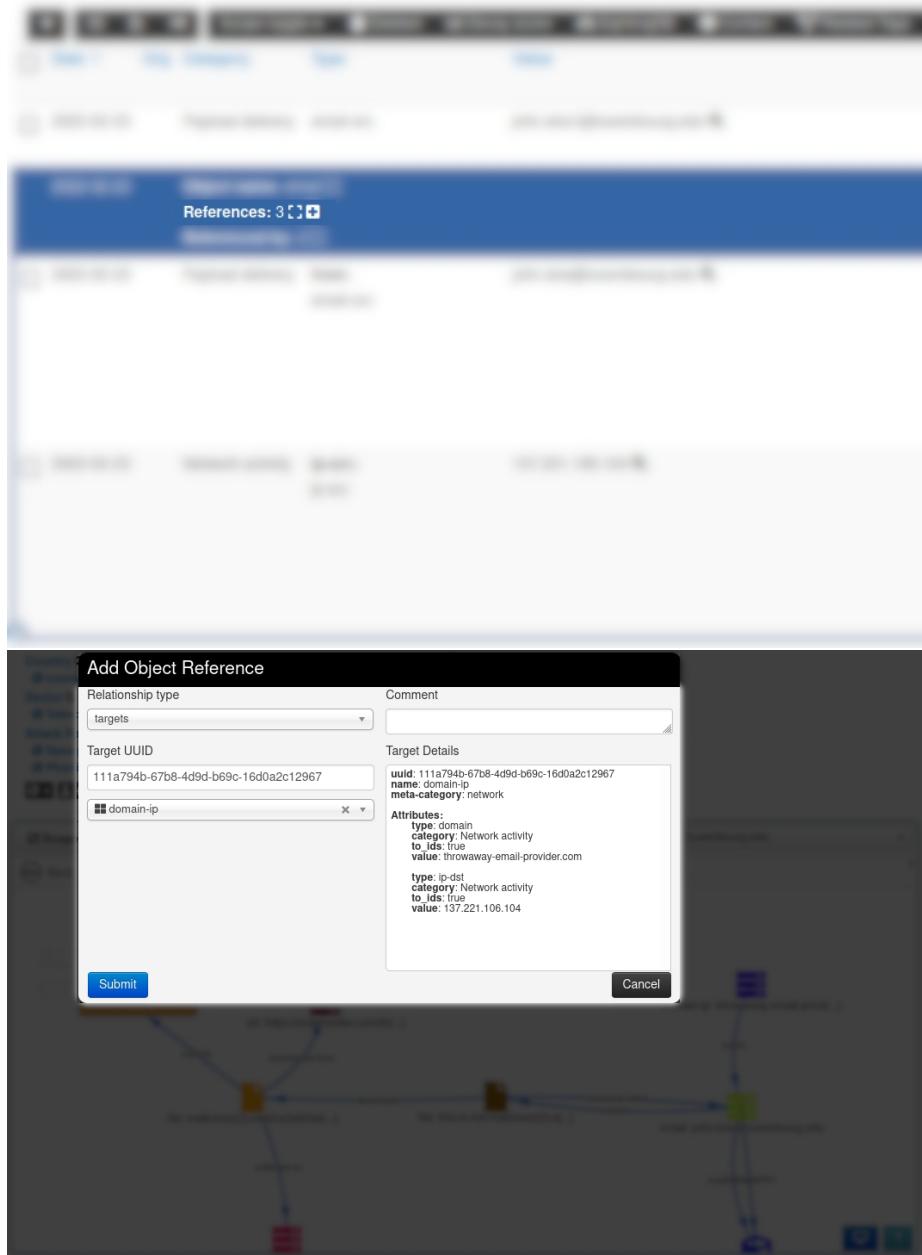
First seen time: Last seen time:
2023-01-01 00:00:00 - 2023-01-01 00:00:00
Expected format: HH:MM:SS:SSS - HH:MM:SS:SSS

Save	Name :: type	Description	Category	Value	IDS	Disable Correlation
<input type="checkbox"/>	Attachment attachment	A non-malicious file.	External analysis	<input type="button" value="Browse..."/> No file selected.	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Malware-sample malware-sample	The file itself (binary)	Payload delivery	<input type="button" value="Browse..."/> No file selected.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Text text	Free text value to attach to the file	Other	<input type="text"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Create an Relationship

1. To create a relationship or *Reference*, a user can either click on the plus button from the *Object* table or do it directly from the *Event Graph*
2. On the *Event Graph*, click on **Edit** then drag an arrow from the first *Object* to another entity
3. On the **Add Object Reference** box, select which verb should be use to describe the relationship in the **Relationship type** input
 - Note: If you want to use a verb not present in the list, use the **custom** entry

4. Click on Submit



Create an Event Report

1. When viewing an *Event*, click on the toggle button **Event reports**
2. Click on **Add Event Report** and enter the name of the report. As its

content can be written with more ease in the dedicated editor, leave it empty and click on **Submit**

3. Once the list has reloaded, click on the *Event Report* that was created, then on the **Edit report** button
4. Write the report in the editor
 - Note: The **Help** button contains documentation about the supported markdown syntax and how to reference *Attributes*, *Objects* and context.
5. Once you are done, click the **Save** button

The screenshot shows the 'Event Reports' interface. At the top, there's a navigation bar with tabs like 'Event reports', 'Logs', 'Incidents', 'Control', and 'Dashboard'. Below the navigation is a search bar and a filter section with buttons for '+ Add Event Report', 'Generate report from Event', 'All', 'Default', and 'Deleted'.

ID	Name	Last update	Distribution	Actions
13	My report	2021-11-25 10:29:34	Your organisation only	
14	Report from - https://www.welivesecurity.com/2021/02/02/kobalos-complex-linux-threat-high-performance-computing-infrastructure (1637836339)	2022-01-18 14:00:50	Inherit event	

Below the table, a modal window is open for 'My report'. It has sections for 'Description', 'Details', and 'Attachments'. The 'Description' section contains a rich text editor toolbar and a preview area showing the report content. The 'Details' section shows the report's ID, name, distribution, and last update. The 'Attachments' section lists files attached to the report.

The main content area of the modal is a code editor with syntax highlighting for Markdown. The code includes:

```

# Description of the incident
The incident took place ...

| Column 1 | Column 2 | Column 3 |
| ----- | ----- | ----- |
| Text | Text | Text |

```
def hello():
 print('Hello World')
```

- A reference to 'filename' Attribute: @attribute1 (94b9aec0-1690-41d4-92b7-1370b2b7c5d9)
- A reference to a 'file' Object @object1(16b535ba-c3f8-4984-9600-961aef943693)

```

On the right side of the modal, there's a sidebar with a title 'Description of the incident' and a note 'The incident took place ...'. It also contains a table with columns 'Column 1', 'Column 2', and 'Column 3', and a code block for the 'hello' function. Below the code block, there are two bullet points:

- A reference to **filename** Attribute: `filename ControlT1573.001Encrypted`
- A reference to a **file** Object `file cf5f24cea4cdb2a222670c6a7b18c966`

Add Tags

1. *Tags* can be attached to both *Events* and *Attributes* with the following buttons:
2. To tag the *Event* or the *Attribute* globally, click on the button with the globe icon
3. Select the *Taxonomy* in which the tag is part of or click on **All Tags**
4. Pick the tag then click on **Submit**

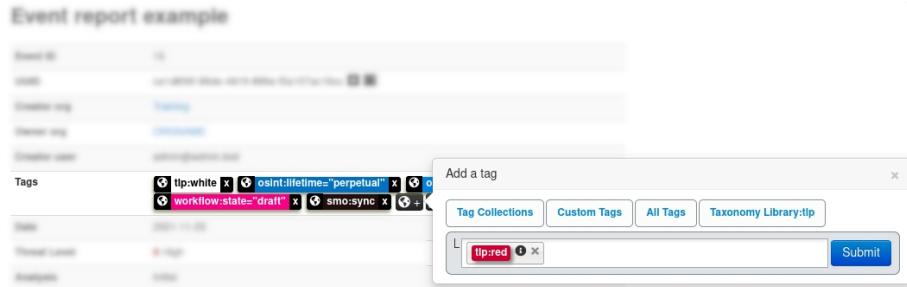


Figure 13: tag

Add Galaxy Clusters

1. Similar to tags, *Galaxy Clusters* can be attached with the button with the globe icon
2. To tag the *Event* or the *Attribute* globally, click on the button with the globe icon
3. Select the namespace in the *Galaxy* is part of or click on All namespaces
4. Select the *Galaxy* in which the *Cluster* is part of or click on All Clusters
5. Pick the *Cluster* then click on Submit

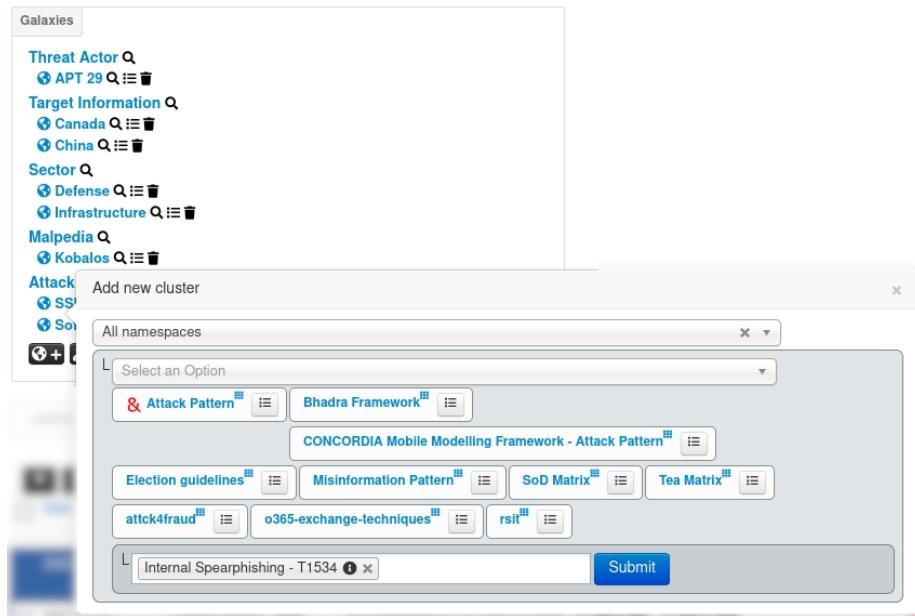


Figure 14: cluster

Publish

1. Whenever an *Event* is to be shared, it has to be Published
2. When viewing an *Event*, click on the Publish button located on the sidebar

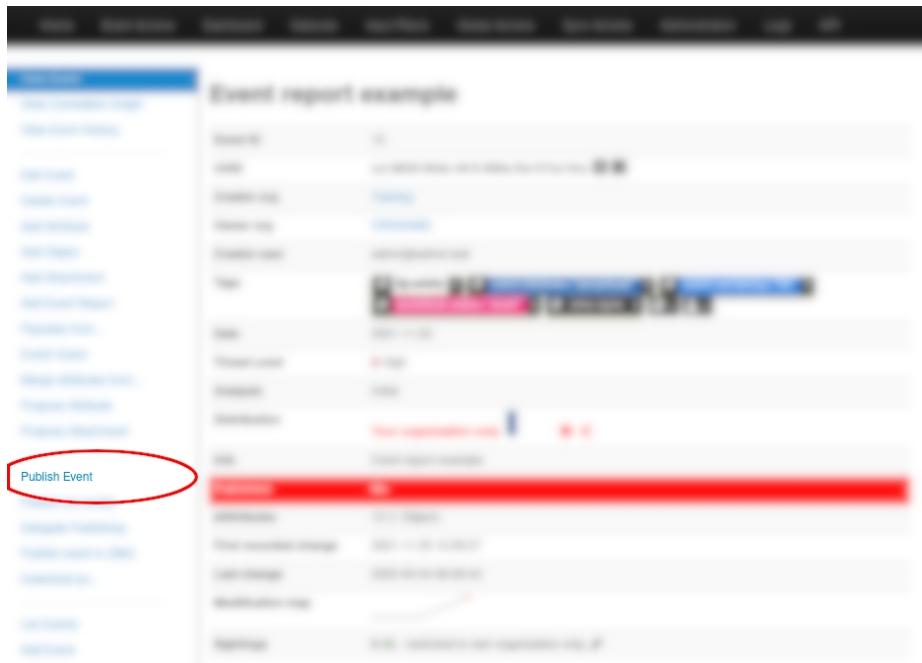


Figure 15: publish-event

Further document and reading references

The following references are not required to follow the training later. The documents and references below are provided for student willing to deep dive into MISP or have specific topic to understand.

- Neolea trainings
- Virtual machines (VirtualBox and VMWare format) if you want to explore a bit more MISP
 - <https://vm.misp-project.org/>
- Slide Deck (source file and compiled)
 - <https://github.com/MISP/misp-training>
 - <https://github.com/cerebrate-project/cerebrate-training>
- Cheatsheet
 - <https://www.misp-project.org/misp-training/cheatsheet.pdf>
- PyMISP

- <https://github.com/MISP/PyMISP/>
- OpenAPI documentation
 - <https://www.misp-project.org/documentation/openapi.html>
- MISP Book
 - User guide for MISP PDF
- MISP data models and knowledge base available
 - MISP taxonomies
 - MISP object templates
 - MISP galaxy