



Report on MISP

CSE 406: Computer Security Sessional

Prepared by:

1805062 - Ayesha Binte Mostofa
1805069 - Sumonta Nandy Amit

Supervised by:

Abdur Rashid Tushar
Lecturer, CSE, BUET

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology

Contents

1	Introduction	3
1.1	Data Sharing	3
1.2	Data Normalization	3
1.3	Threat Intelligence Feeds	3
1.4	Event Correlation	4
1.5	Flexible Taxonomies	4
1.6	Collaboration	4
1.7	Indicators of Compromise (IOCs)	4
1.8	Incident Response	4
1.9	API and Automation	4
1.10	Customization	4
1.11	Privacy and Access Control	4
2	Source Code Overview	5
2.1	MISP core	5
2.2	PyMISP	11
2.3	misp-taxonomies	13
2.4	misp-galaxy	13
2.5	misp-warninglists	14
3	Installation Guide	15
3.1	Resources	15
3.2	Possible errors	15
3.3	Launching MISP in ubuntu and vm	15
3.4	Launching MISP in Docker	15
3.5	Forget password issue resolve	15
4	Features	16
4.1	Add events	16
4.2	Data modelling	16
4.3	Data feeds	17
4.4	Sharing and Collaboration	17
4.5	Block Events	18
4.6	Customized tag creation and usage	18
4.7	Adding attributes automatically	19
4.8	Event Graph Generation	21
4.9	Generation of Event Report	23
4.10	MISP API	25
4.11	Import and Export events	30
4.12	Taxonomy	30
4.13	Galaxy	31
5	unlisted youtube video link created by us	32

1 Introduction

MISP (Malware Information Sharing Platform & Threat Sharing) is an open-source platform designed for sharing, analyzing, and collaborating on structured threat information within the cybersecurity community. It offers several key features:

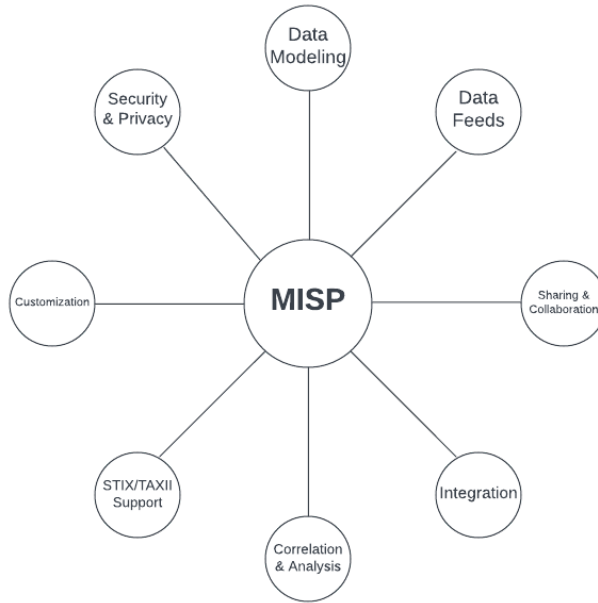


Figure 1: Key Features of MISP

1.1 Data Sharing

MISP facilitates the sharing of cybersecurity threat data among organizations, enhancing the collective ability to respond to evolving threats.

1.2 Data Normalization

The platform enforces a common structure for describing and normalizing threat information, ensuring consistency in data representation and analysis.

1.3 Threat Intelligence Feeds

Users can integrate external threat intelligence feeds, enriching their data with up-to-date information from reputable sources.

1.4 Event Correlation

MISP allows the correlation of related events, providing a contextual understanding of threats and their potential impact.

1.5 Flexible Taxonomies

The platform supports various taxonomies and classification systems, enabling organizations to categorize and label threat information as needed.

1.6 Collaboration

MISP encourages collaboration between organizations, facilitating the sharing of knowledge and expertise to collectively defend against cyber threats.

1.7 Indicators of Compromise (IOCs)

Users can share IOCs like IP addresses, domain names, hashes, and other artifacts to identify and mitigate threats.

1.8 Incident Response

MISP assists in incident response by centralizing information and providing a structured way to analyze and share data related to ongoing security incidents.

1.9 API and Automation

The platform provides an API for task automation, simplifying integration into existing security workflows and tools.

1.10 Customization

Organizations can customize MISP to match their needs by adapting taxonomies, attributes, and configurations.

1.11 Privacy and Access Control

MISP offers granular data access control, allowing organizations to define who can access, share, and contribute information.

2 Source Code Overview

There are a total of 82 repositories under the MISP project. The most popular ones are discussed here. Most of the programs are written in Python.

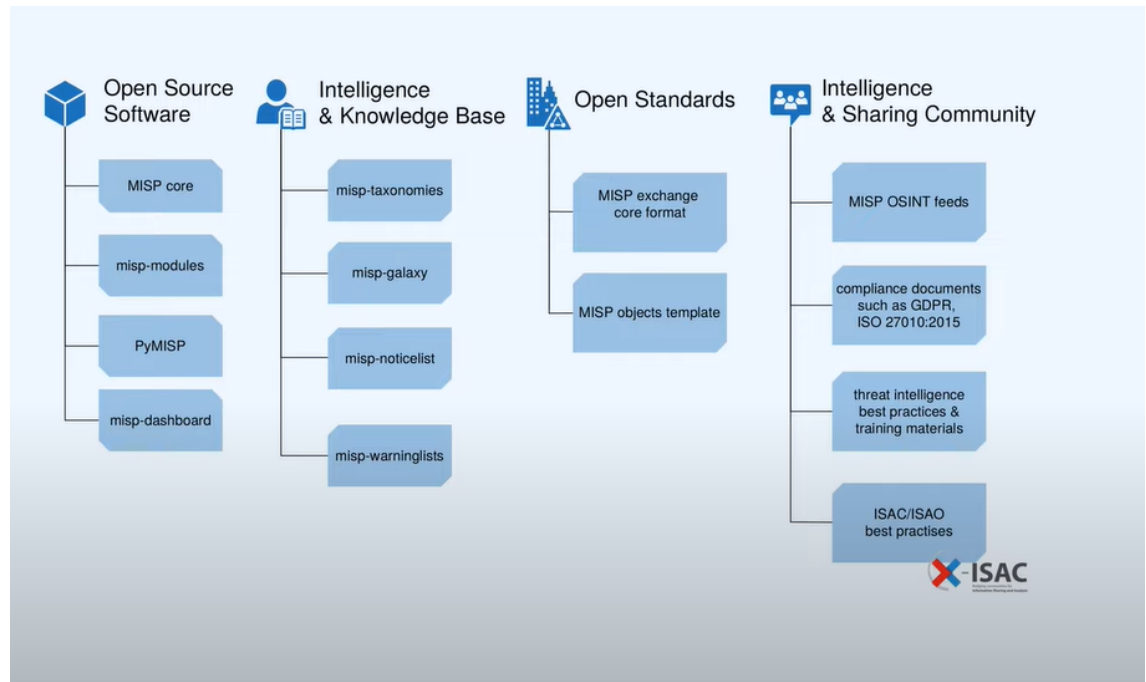


Figure 2: MISP Project

2.1 MISP core

This is the main module of the MISP project. starts with app/index.php

```
1  <?PHP
2  ...
3  require 'webroot' . DIRECTORY_SEPARATOR . 'index.php';

app/Model/Event.php

1  <?php
2  App::uses('AppModel', 'Model');
3  App::uses('CakeEmail', 'Network/Email');
4  App::uses('AttachmentTool', 'Tools');
5  App::uses('TmpFileTool', 'Tools');
6  App::uses('SendEmailTemplate', 'Tools');
7  App::uses('ProcessTool', 'Tools');
8  ...
9  class Event extends AppModel
10 {
```

```

11     ...
12     private function __attachAttributeTags(array &$events,
13         $excludeLocalTags = false)
14     ...
15         private function __attachTags(array &$event, $justExportable)
16     ...
17     public function restSearch(array $user, $returnFormat, $filters
18         , $paramsOnly = false, $jobId = false, &$elementCounter = 0, &
19         $renderView = false)
20     {
21     ...
22         public function clusterEventIds($exportTool, $eventIds)
23     ...
24     public function add_original_file($file, $original_filename,
25         $event_id, $format)
26     ...
27     private function getRequiredTaxonomies()
28     {
29         $this->Taxonomy = ClassRegistry::init('Taxonomy');
30         return $this->Taxonomy->find('column', array(
31             'conditions' => array('Taxonomy.required' => 1, '
32             Taxonomy.enabled' => 1),
33             'fields' => array('Taxonomy.namespace')
34         ));
35     }
36     ...
37     public function extractAllTagNames(array $event)
38     {
39         $tags = array();
40         if (!empty($event['EventTag'])) {
41             foreach ($event['EventTag'] as $eventTag) {
42                 $tagName = $eventTag['Tag']['name'];
43                 $tags[$tagName] = $tagName;
44             }
45         }
46     }
47     ...
48     public function getExtendingEventIdsFromEvent($user, $eventID)
49     ...
50     public function getEventRepublishBanStatus($eventID)
51     {
52         $banStatus = [
53             'error' => false,
54             'active' => false,
55             'message' => __('Event publish is not banned')
56         ];
57     }
58     ...
59     public function exportTypes()
60     {
61         return array(
62             'json' => array(
63                 'extension' => '.json',
64                 'type' => 'JSON',
65                 'scope' => 'Event',
66                 'requiresPublished' => 0,
67                 'params' => array('includeAttachments' => 1, '
68                 ignore' => 1, 'returnFormat' => 'json'),
69                 'description' => __('Click this to download all

```

```

        events and attributes that you have access to in MISP JSON
        format.'),
62         ),
63     ...
64     public function publishEventToZmq($id, $user, &$fullEvent)
65     ...
66     public function publishEventToKafka($id, $user, &$fullEvent,
        $kafkaTopic)
67     ...
68     public function getTrendsForTags(array $user, array
        $eventFilters=[], int $baseDayRange, int $rollingWindows=3,
        $tagFilterPrefixes=null): array
69     ...
70     public function getTrendsForTagsFromEvents(array $events, int
        $baseDayRange, int $rollingWindows=3, $tagFilterPrefixes=null):
        array
71     ...
72     public function extractRelatedCourseOfActions(array $events):
        array
73     ...

```

app/Model/Attribute.php

```

1     class Attribute extends AppModel
2     {
3     ...
4     const EDITABLE_FIELDS = [
5         'timestamp',
6         'category',
7         'value',
8         'value1',
9     ...

```

app/Model/EventGraph.php

```

1     <?php
2     App::uses('AppModel', 'Model');
3
4     class EventGraph extends AppModel
5     {
6     ...
7     public function getPictureData($eventGraph)
8     {
9     ...

```

app/Model/EventReport.php

```

1     <?php
2     App::uses('AppModel', 'Model');
3     ...
4     class EventReport extends AppModel
5     {
6     ...
7     public function captureReport(array $user, array $report,
        $eventId)
8     {
9     ...
10    public function editReport(array $user, array $report,
        $eventId, $fromPull = false, &$nothingToChange = false)

```



```

11     {
12     ...
13     public function deleteReport(array $user, $report, $hard=false)
14     {
15     ...

```

app/Model/EventBlocklist.php

```

1  <?php
2  App::uses('AppModel', 'Model');
3
4  class EventBlocklist extends AppModel
5  {
6  ...
7  public function isBlocked($eventUuid)
8  {
9  ...
10 public function removeBlockedEvents(array &$eventArray)
11 {
12 ...

```

app/Model/Feed.php

```

1  <?php
2  App::uses('AppModel', 'Model');
3  App::uses('RandomTool', 'Tools');
4  App::uses('TmpFileTool', 'Tools');
5  App::uses('AttributeValidationTool', 'Tools');
6
7  class Feed extends AppModel
8  {
9  ...
10 private function getCachedFeedsOrServers(array $user, $scope)
11 {
12 ...
13 private function downloadFromFeed(array $actions, array $feed,
14   HttpSocket $HttpSocket = null, array $user, $jobId = false)
15 {
16 ...
17 private function __createFeedRequest($headers = false)
18 {
19 ...

```

app/Model/Galaxy.php

```

1  <?php
2  App::uses('AppModel', 'Model');
3
4  /**
5   * @property GalaxyCluster $GalaxyCluster
6   * @property Galaxy $Galaxy
7   */
8  class Galaxy extends AppModel
9  {
10 ...
11 private function __load_galaxies($force = false)
12 {
13 ...
14 private function __getPreExistingClusters(array $galaxies,
15   array $cluster_package)

```

```

14     {
15     ...
16     private function __createClusters($cluster_package, $template)
17     {
18     ...
19     public function captureGalaxy(array $user, array $galaxy)
20     {
21     ...
22     public function importGalaxyAndClusters(array $user, array
23     $clusters)
24     {
25     ...

```

app/Model/Job.php

```

1  <?php
2  App::uses('AppModel', 'Model');
3
4  class Job extends AppModel
5  {
6  ...
7  public function createJob($user, $worker, $jobType, $jobInput,
8  $message = '')
9  {
10 ...
11 public function saveProgress($jobId = null, $message = null,
12 $progress = null)
13 {
14 ...
15 public function saveStatus($jobId = null, $success = true,
16 $message = null)
17 {
18 ...

```

app/Model/MispObject.php

```

1  <?php
2  App::uses('AppModel', 'Model');
3  App::uses('TmpFileTool', 'Tools');
4  App::uses('AttributeValidationTool', 'Tools');
5  App::uses('FileAccessTool', 'Tools');
6
7  /**
8   * @property Event $Event
9   * @property SharingGroup $SharingGroup
10  * @property Attribute $Attribute
11  * @property ObjectReference $ObjectReference
12  * @property ObjectTemplate $ObjectTemplate
13  */
14  class MispObject extends AppModel
15  {
16  ...
17  $simple_params = array(
18      'Object' => array(
19          'object_name' => array('function' => '
20  set_filter_object_name'),
21          'object_template_uuid' => array('function' => '
22  set_filter_object_template_uuid'),

```

```

21         'object_template_version' => array('function'
=> 'set_filter_object_template_version'),
22         'deleted' => array('function' => '
set_filter_deleted')
23     ),
24     ...
25
26     private function checkForDuplicateObjects($object, $eventId, &
$duplicatedObjectId, &$duplicateObjectUuid)
27     {
28         ...
29         public function saveObject(array $object, $eventId, $template
= false, array $user, $errorBehaviour = 'drop',
$breakOnDuplicate = false)
30     {
31         ...
32         public function deltaMerge(array $object, array $objectToSave,
$onlyAddNewAttribute=false, array $user)
33     {
34         ...

```

app/Model/Tag.php

```

1  <?php
2  App::uses('AppModel', 'Model');
3
4  /**
5   * @property EventTag $EventTag
6   * @property User $User
7   * @property AttributeTag $AttributeTag
8   * @property FavouriteTag $FavouriteTag
9   * @property Organisation $Organisation
10  */
11  class Tag extends AppModel
12  {
13      ...
14      public function lookupTagIdForUser(array $user, $tagName)
15      {
16          ...
17      public function fetchUsableTags(array $user, $isGalaxy = null)
18      {
19          ...

```

app/Model/Taxonomy.php

```

1  <?php
2  App::uses('AppModel', 'Model');
3
4  /**
5   * @property TaxonomyPredicate $TaxonomyPredicate
6   */
7  class Taxonomy extends AppModel
8  {
9      ...
10     private function __getTaxonomy($id, $filter = false)
11     {
12         ...
13     public function getAllTaxonomyTags($inverse = false, $user =
false, $full = false, $hideUnselectable = true, $local_tag =

```

```

14     false)
15     {
16     ...
16     public function getTaxonomyTags($id, $upperCase = false,
16     $existingOnly = false)
17     {
18     ...
19     public function addTags($id, $tagList = false)
20     {
21     ...
22     public function checkIfNewTagIsAllowedByTaxonomy($newTagName,
22     array $tagNameList=array())
23     {
24     ...

```

2.2 PyMISP

Repository Link: <https://github.com/MISP/PyMISP/> PyMISP is a Python library to access MISP platforms via their REST API. PyMISP allows you to fetch events, add or update events/attributes, add or update samples, or search for attributes.

The functionalities were used when we integrated the MISP instance with the Cortex in our demo.

Dependencies:

- Python 3.10
- fileobjects: to create PE/ELF/Mach-o objects
- openioc: to import files in OpenIOC format (not really maintained)
- virustotal: to query VirusTotal and generate the appropriate objects
- docs: to generate the documentation
- pdfexport: to generate PDF reports out of MISP events
- url: to generate URL objects out of URLs with Pyfaup
- email: to generate MISP Email objects
- brotli: to use the brotli compression when interacting with a MISP instance

PyMISP classes:

- PyMISP: This class is responsible for processing the URL of the MISP instance you want to connect to and The API key of the user you want to use
- MISPAbstract

- MISPEncode
- MISPEvent
- MISPEventBlocklist
- MISPEventDelegation
- MISPAtribute
- MISPObject
- MISPObjectAttribute
- MISPObjectReference
- MISPObjectTemplate
- MISPTag
- MISPUser
- MISPUserSetting
- MISPOrganisation
- MISPOrganisationBlocklist
- MISPFee
- MISPIinbox
- MISPLog
- MISPNoticelist
- MISPRole
- MISPServer
- MISPShadowAttribute
- MISPSharingGroup
- MISPSighting
- MISPTaxonomy
- MISPWarninglist

PyMISP Tools

- File Object
- ELF Object

- PE Object
- Mach-O Object
- VT Report Object
- STIX
- OpenIOC

How to use PyMISP: We can find scripts and examples in the example directory

```
1 cd examples
2 cp keys.py.sample keys.py
3 vim keys.py
```

2.3 misp-taxonomies

Repository Link: <https://github.com/MISP/misp-taxonomies> MISP Taxonomies is a set of common classification libraries to tag, classify, and organize information. Taxonomy allows to express same vocabulary among a distributed set of users and organizations.

Available taxonomies: There are a lot of built-in taxonomies available in this directory. e.g.: CERT-XLM, DFRLab-dichotomies-of-disinformation, Gray-Zone etc.

Taxonomy structure A JSON file describing taxonomy as triple tags inside a directory-matching namespace.

How to add private taxonomy?

```
1 cd /var/www/MISP/app/files/taxonomies/
2 mkdir privatetaxonomy
3 cd privatetaxonomy
4 vi machinetag.json
```

2.4 misp-galaxy

Repository Link: <https://github.com/MISP/misp-galaxy> MISP galaxy is a simple method to express a large object called a cluster that can be attached to MISP events or attributes. A cluster can be composed of one or more elements. Elements are expressed as key values. There are default knowledge bases (such as Threat Actors, Tools, Ransomware, ATT & CK matrixes) available in MISP galaxy but those can be overwritten, replaced, updated, forked, and shared. Some available galaxies are 360.net Threat Actors, Android, Azure Threat Research Matrix, etc.

Testing the galaxies:

```

1  sudo apt install jq moreutils python3-jsonschema
2  sudo wget -O /usr/local/bin/jsonschema https://gist.
   githubusercontent.com/SteveClement/
   e6ac60e153e9657913000216fc77c6ef/raw/
   c273ace06ad338d609dd2c84a0a6e215a268ea11/jsonschema
3  sudo chmod +x /usr/local/bin/jsonschema # This will only work
   with jsonschema >2.4 (before no CLI interface was available)

```

2.5 misp-warninglists

Repository link: <https://github.com/MISP/misp-warninglists>

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

warning list format:

```

1  {
2  "name": "List of known public DNS resolvers",
3  "version": 1,
4  "description": "Event contains one or more public DNS resolvers
   as attribute with an IDS flag set",
5  "matching_attributes": [
6    "ip-src",
7    "ip-dst"
8  ],
9  "list": [
10    "8.8.8.8",
11    "8.8.4.4",
12    "208.67.222.222",
13    "208.67.220.220",
14    "195.46.39.39",
15    "195.46.39.40"
16  ]
17 }

```

types of warning list

- string
- substring
- hostname
- cidr

3 Installation Guide

3.1 Resources

- Installation in VM and Ubuntu: <https://www.youtube.com/watch?v=nZcTc60YsIs>
Documentation: <https://misp.github.io/MISP/INSTALL.ubuntu2004>
- Installation using Docker: https://youtu.be/h_LxGcvjg8U

3.2 Possible errors

- **ERROR:** cannot verify raw.githubusercontent.com's certificate, issued by 'CN=DigiCert TLS RSA SHA256 2020 CA1,O=DigiCert Inc,C=US':

Sol ⁿ:

- create a `~/.wgetrc` file
- add `ca_certificate=/etc/ssl/certs/ca-certificates.crt` in it

3.3 Launching MISP in ubuntu and vm

- open shell
- find 'inet' for enp0s3 value using 'ifconfig'
- open browser and put the 'inet' value in the address bar
- give the credentials

```
1      Email: admin@admin.test
2      Password: admin
3
```

- change the password

3.4 Launching MISP in Docker

- Change Localhost:8080 in docker-compose.yml file.
- Login using admin@admin.test email and pass
- change the password

3.5 Forget password issue resolve

- Use this command line : `sudo /var/www/MISP/app/Console/cake Password email newpassword`

4 Features

4.1 Add events

- A community can share threat events among the community

The screenshot shows the 'Add Event' interface. On the left is a sidebar with navigation links: List Events, Add Event (highlighted), Import from..., REST client, List Attributes, Search Attributes, View Proposals, Events with proposals, View delegation requests, View periodic summary, Export, and Automation. The main form has the following fields: Date (2023-09-13), Distribution (This community only), Threat Level (High), Analysis (Initial), Event info (test), and Extends Event (Event UUID or ID. Leave blank if not applicable.). A blue 'Submit' button is at the bottom.

4.2 Data modelling

- MISP uses a flexible data model that allows users to define and share threat intelligence information in a structured way. This includes information about malware, indicators of compromise (IoCs), attack techniques, and more. Attributes and object types, Event Structure and Data sharing formats can be selected from various options in MISP.
- Add attribute options in adding events

The screenshot shows the 'Add Attribute' form. It has two dropdown menus: 'Category' and 'Type'. The 'Category' dropdown is open, showing options like Internal reference, Targeting data, Antivirus detection, Payload delivery, Artifacts dropped, Payload installation, Persistence mechanism, Network activity, and Payload type. The 'Type' dropdown is also open, showing options like md5, sha1, sha256, filename, pdb, filename:md5, filename:sha1, filename:sha256, and ip:src. Below the dropdowns are checkboxes for 'Batch import', 'For Intrusion Detection System', and 'Disable Correlation'. There are also fields for 'First seen date', 'Last seen date', 'First seen time', and 'Last seen time'.

- Add object options in adding events

The screenshot shows the 'Add Object' form. It has a 'Select an Option' dropdown menu. The dropdown is open, showing options like All Objects, attribute, file, financial, followthemoney, health, internal, iot, malware, and misc. The 'All Objects' option is selected.

4.3 Data feeds

- Users in the community can see the news and threat events whenever they log in.
- Event feed

← previous

next →

☒

My Events

☒

Org Events

Event info

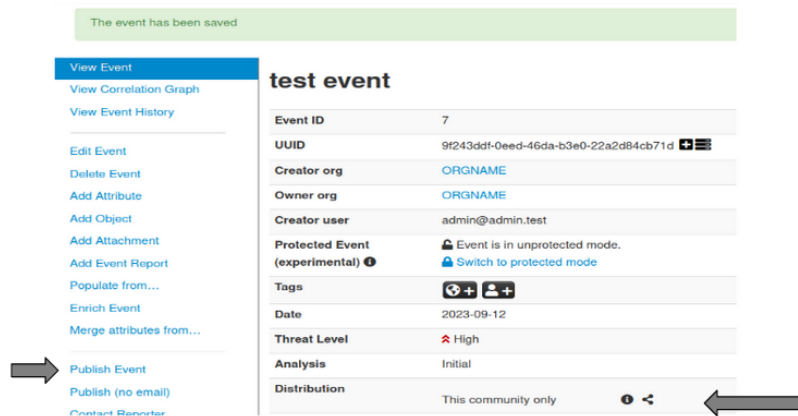
Filter

<input checked="" type="checkbox"/>	Creator org	Owner org	ID	Clusters	#Att.	#Conn.	Creator user	Date	Last modified at	Published at	Info	Distribution	Actions
<input checked="" type="checkbox"/>	X	ORGNAME	ORGNAME	A.7	0		admin@admin.test	2023-09-12	2023-09-12 14:04:50		test event	Community <	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	X	ORGNAME	ORGNAME	A.6	0		admin@admin.test	2023-09-12	2023-09-12 12:54:27		Test publishing	Community <	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	X	ORGNAME	ORGNAME	A.5	0		ayeshashok@gmail.com	2023-09-12	2023-09-12 12:44:18		Test publishing	Community <	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	✓	ORGNAME	ORGNAME	A.4	360.net Threat Actors <input checked="" type="checkbox"/>		admin@admin.test	2023-09-12	2023-09-12 09:17:38	2023-09-12 09:18:17	Firewall blocking event	Community <	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	X	ORGNAME	ORGNAME	A.3	0		admin@admin.test	2023-09-12	2023-09-12 09:13:20		Firewall blocking event	Community <	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	✓	ORGNAME	ORGNAME	A.2	1		ayeshashok@gmail.com	2023-08-30	2023-08-30 16:30:49	2023-08-30 16:31:25	firewall dom	Organisation <	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	✓	ORGNAME	ORGNAME	A.1	11	1	admin@admin.test	2023-08-30	2023-08-30 15:55:39	2023-08-30 15:56:55	test Malware	Community <	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

- News feed

4.4 Sharing and Collaboration

- When the admin publishes the event among the users, it is shared among the community



4.5 Block Events

- If an event is undesirable, it can be blocked

Add Event Blocklist Entries

Simply paste a list of all the event UUIDs that you wish to block from being entered.
UUIDs

Enter a single or a list of UUIDs

Creating organisation
(Optional) The organisation that the event is associated with

Event info
(Optional) the event info of the event that you would like to block. It's best to leave this empty if you are adding a list of UUIDs.

Comment
(Optional) Any comments you would like to add regarding this (or these) entries.



4.6 Customized tag creation and usage

- Tag can be created and can be used in adding events

Add Tag

Name

Colour

Restrict tagging to org

Restrict tagging to user

☒ Exportable
☐ Hide Tag
☐ Enforce this tag to be used as local only

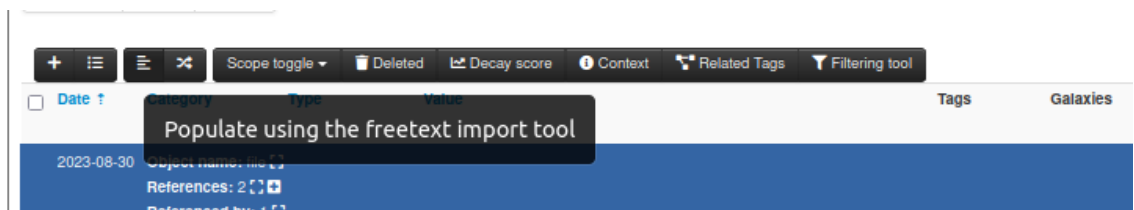
testing created tag

Event ID	8
UUID	f2da80da-1c27-4a24-94ca-3cf7615aced5
Creator org	ORGNAME
Owner org	ORGNAME
Creator user	admin@admin.test
Protected Event (experimental)	Event is in unprotected mode. <input type="checkbox"/> Switch Add a tag
Tags	<input type="button" value="Tag Collections"/> <input type="button" value="Custom Tags"/> <input type="button" value="All Tags"/>
Date	2023-09
Threat Level	High
Analysis	Initial
Distribution	

4.7 Adding attributes automatically

This helps to find the attributes of a given text. The user does not need to add attributes manually for every event.

- select an event to add attributes
- Select 'populate using the free text import tool'



Fretext Import Tool

Paste a list of IOCs into the field below for automatic detection.

https://evilprovider.com/this-is-not-malicious.exe (also attached, resolves to 2607:5300:60:cd52:304b:760d:da7:d5). It looks like the sample is trying to exploit CVE-2015-5465. After a brief triage, the secondary payload has a hardcoded C2 at https://another.evil.provider.com:57666 (118.217.182.36) to which it tries to exfiltrate local credentials. This is how far we have gotten so far. Please be mindful that this is an ongoing investigation, we would like to avoid informing the attacker of the detection and kindly ask you to only use the contained information to protect your constituents.

Submit
Cancel

- submit the text to encode
- submit attributes

Fretext Import Results

Below you can see the attributes that are to be created. Make sure that the categories and the types are correct, often several options will be offered based on an inconclusive automatic resolution.

☐ Proposals instead of attributes

Value	Similar Attributes	Category	Type	IDS	Disable Correlation	Distribution	Comment
john.doe@luxembourg.edu	1643	Payload delivery	email-src	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Inherit event	
throwaway-email-provider.com	1643	Network activity	domain	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Inherit event	
137.221.106.104	1643	Network activity	ip-dst	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Inherit event	
https://evilprovider.com/this-is-not-malicious.exe	1643	Network activity	url	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Inherit event	
2607:5300:60:cd52:304b:760d:da7:d5	1643	Network activity	ip-dst	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Inherit event	
CVE-2015-5465	1643	External analysis	vulnerability	<input type="checkbox"/>	<input type="checkbox"/>	Inherit event	
https://another.evil.provider.com		Network activity	url	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Inherit event	On port 57666
118.217.182.36	1643	Network activity	ip-dst	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Inherit event	

Submit attributes

email → email-src Change all

Update all comment fields Change all

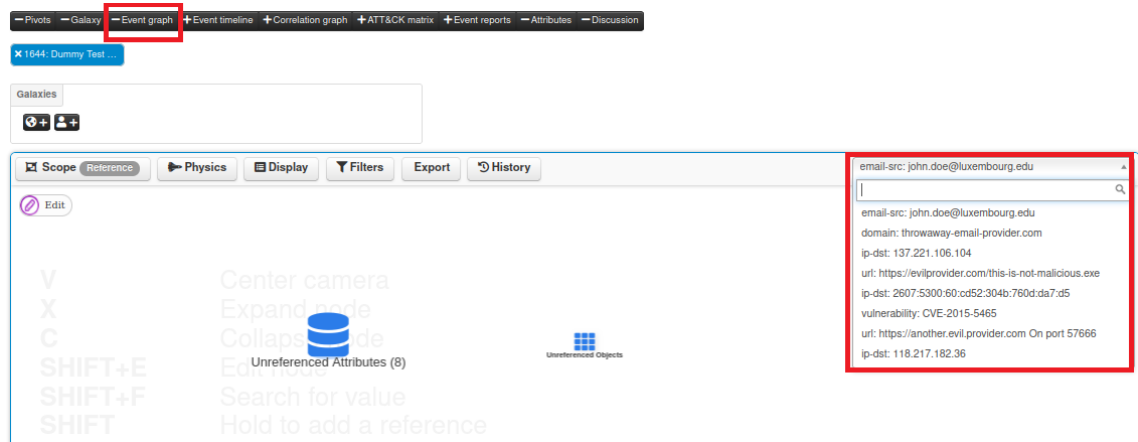
- Attributes will then be added automatically

2023-08-30 Object name: person[]		References: 3				Inherit			
<input type="checkbox"/>	2023-08-25 Other	function:	Teacher of CEO's daughter	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2023-08-25 Person	last-name:	Doe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2023-08-25 Person	full-name:	John Doe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2023-08-25 Person	first-name:	John	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2023-08-25 Payload delivery	e-mail:	john.doe@luxembourg.edu	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2023-08-25 Other	role:	Victim	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

4.8 Event Graph Generation

An event graph helps to visualize the flow of incidents.

- select an event to generate an event graph
- Select the event graph option and load all attributes



- The reference between objects and attributes can be added using the 'Edit & Add Reference' option

Add Object Reference

Relationship type

custom

Comment

Target UUID

47d4c143-d1ec-4646-80dd-25b9444e2fea

Target Details

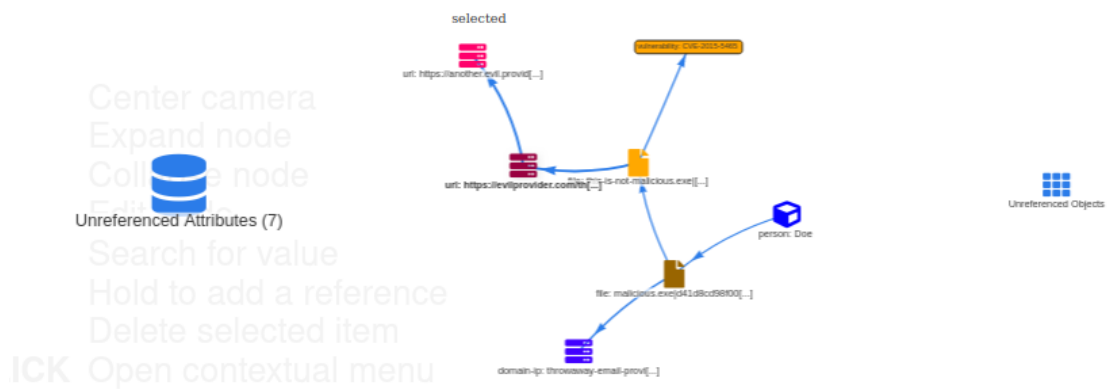
Select an Option

extriterates to

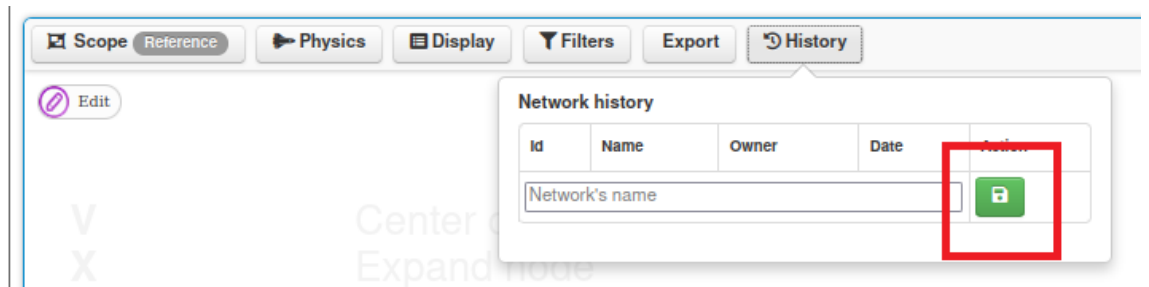
Submit

Cancel

- Finally, the event graph will look like this



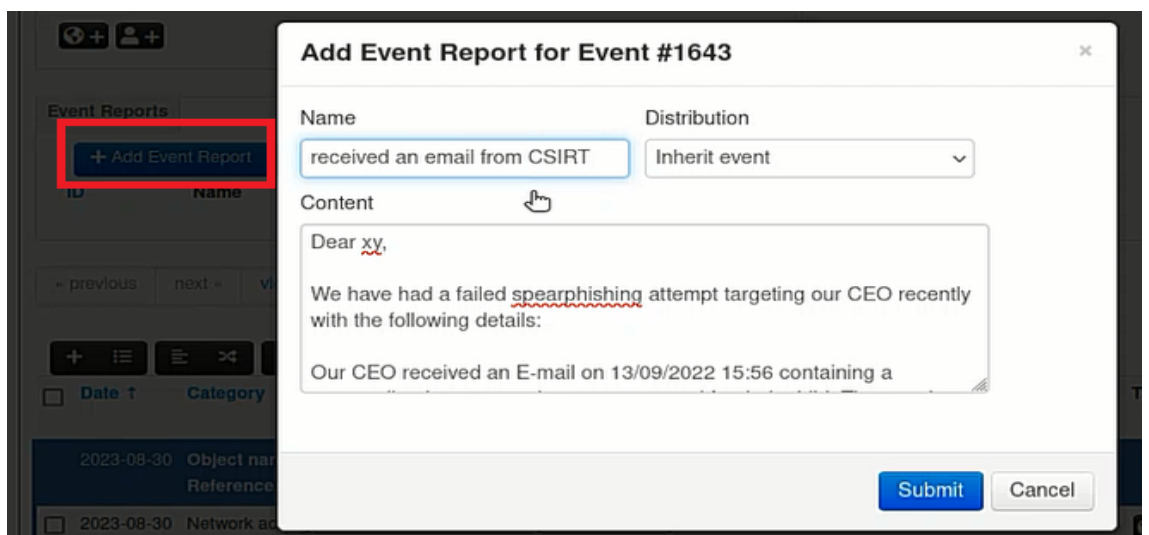
- The network should be saved to be used later



4.9 Generation of Event Report

A documentation of an event.

- Create an event report



- Edit the report, adding the encoded entities in the report

Event report: received an email from CSIRT

Markdown **Raw** **Edit report**

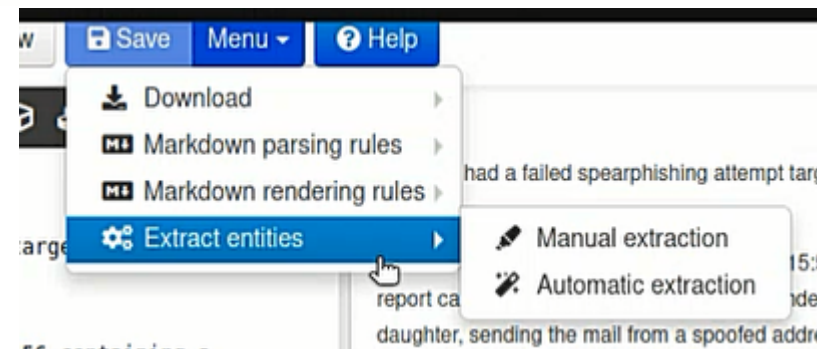
Dear xy,

We have had a failed spearphishing attempt targeting our CEO recently with the following details:

Our CEO received an E-mail on 13/09/2022 15:56 containing a personalised message about a report card for their child. The email was received from a spoofed address (john.doe@luxembourg.edu). John Doe is a teacher of the student. The email was received from

The e-mail contained a malicious file (find it attached) that would try to download a secondary payload from <https://evilprovider2607:5300:60:cd52:304b:760d:da7:d5>. It looks like the sample is trying to exploit CVE-2015-5465. After a brief triage, the secondary payload (118.217.182.36) to which it tries to exfiltrate local credentials. This is how far we have gotten so far. Please be mindful that this is a detection and kindly ask you to only use the contained information to protect your constituents.

Best regards,



Data Replacement19

Context replacement0

Fullscreen

Data extraction0

Value	Existing attribute	Occurrences	Action
0	[Other] size-in-bytes 480477	11	Repla
57666	[Network activity] port 480493	1	Repla
CVE-2015-5465	[External analysis] vulnerability 480489	1	Repla
Doe	[Person] last-name 480465	1	Repla
John Doe	[Person] full-name 480466	1	Repla
John	[Person] first-name 480467	1	Repla
john.doe@luxembourg.edu	[Payload delivery] email-src 480468	1	Repla
throwaway-email-provider.com	[Network activity] domain 480470	1	Repla
137.221.106.104	[Network activity] ip-dst 480471	1	Repla
malicious.exe	[Payload delivery] filename 480473	1	Repla

report card for their child. The attacker pretended to be working for the school of th daughter, sending the mail from a spoofed address (john.doe@luxembourg.edu). J teacher of the student. The email was received from

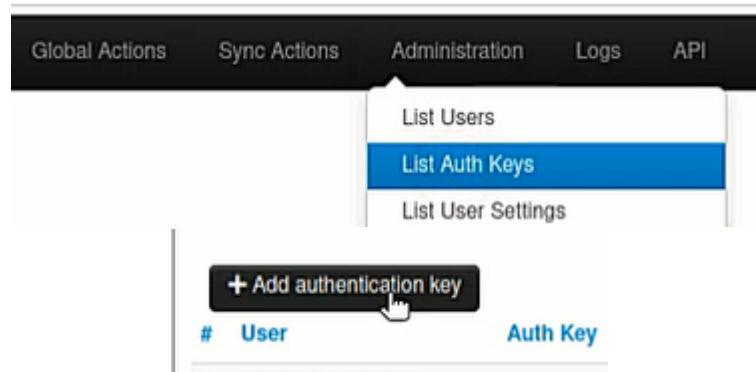
domain-ip → domain throwaway-email-provider.com (137.221.106.104).

The e-mail contained a malicious file (find it attached) that would try to download a

4.10 MISP API

We can integrate the MISP database with any analyzer through its API feature. Here, we will incorporate the MISP instance with Cortex. At first, we create the API authentication key.

- From administration, select List Auth keys

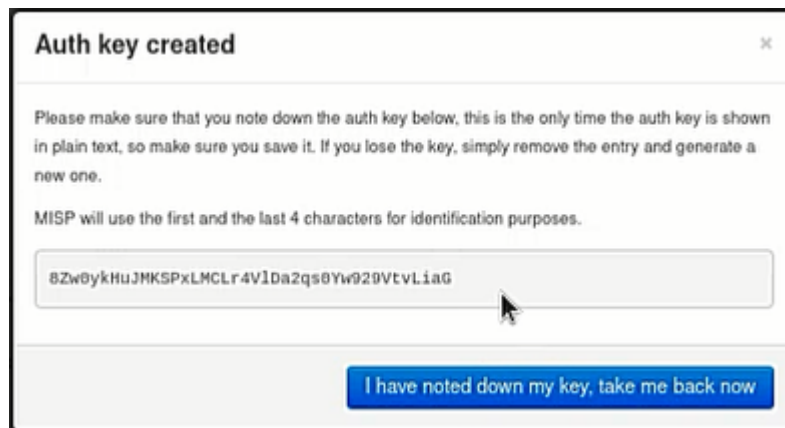


- Fill in the data as required.

A screenshot of a form titled 'Add auth key'. The form contains the following fields and options:

- A text area for 'Auth keys are used for API access. A user can have more than one authkey, so if you would like to use separate keys per tool that queries MISP, add additional keys. Use the comment field to make identifying your keys easier.'
- A dropdown menu for 'User' with the value 'admin@admin.test'.
- A text area for 'Comment'.
- A text area for 'Allowed IPs' with the value '0.0.0.0/0'.
- A text input for 'Expiration (keep empty for indefinite)' with the value '2023-12-31'.
- A checkbox labeled 'Read only (it will be not possible to do any change operation with this token)'.
- At the bottom right, there are two buttons: 'Submit' and 'Cancel'.

- Save the generated Auth key, as we cannot access it later.

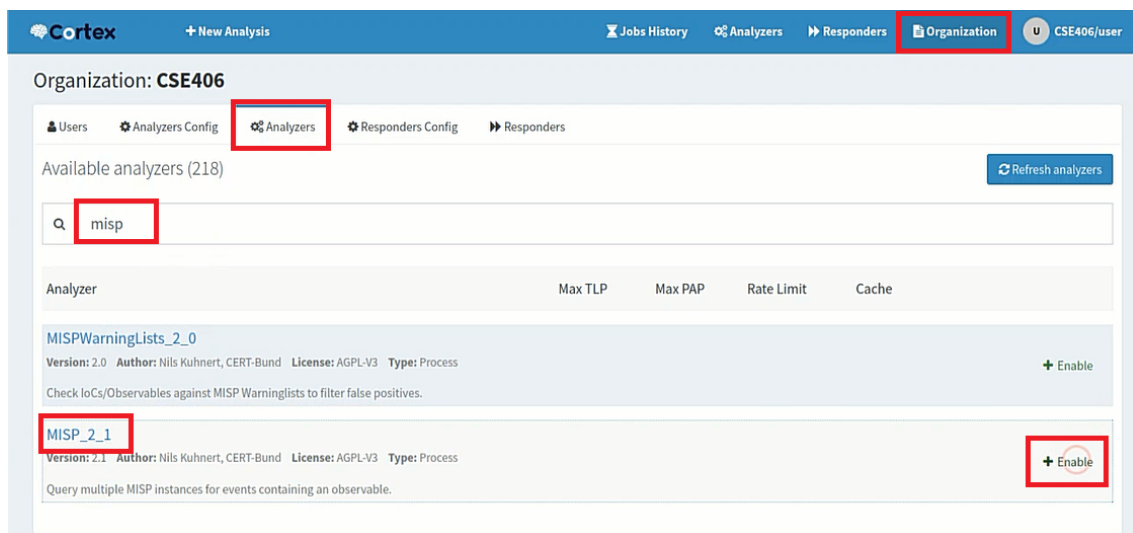


- The auth key is listed and ready to share.

#	User	Auth Key	Expiration	Last used
1	admin@admin.test	xvys*****XGxB	Indefinite	Never
2	admin@admin.test	8Zw8*****LiaG	2023-12-31 00:00:00	Never

Now, We incorporate this with Cortex.

- log into the cortex.
- add the MISP instance



Enable analyzer MISP_2_1

Base details

Name

Configuration [Apply defaults](#)

name [Add option](#)

1. [✕](#)

Name of MISP servers

url * [Add option](#)

1. [✕](#)

URL of MISP servers

key * [Add option](#)

1. [✕](#)

API key for each server

cert_check *

Verify server certificate

- MISP instance added

Analyzers (1)

Data Types (13) **Analyzer** **Page size**

Select [Search](#) [Clear](#) 50 / page

MISP_2_1 **Version:** 2.1 **Author:** Nils Kuhnert, CERT-Bund **License:** AGPL-V3

Query multiple MISP instances for events containing an observable. [Run](#)

Applies to: [domain](#) [ip](#) [url](#) [fqdn](#) [uri_path](#) [user-agent](#) [hash](#) [mail](#) [mail_subject](#) [registry](#) [regex](#) [other](#) [filename](#)

How to use it?

- Now we want to search for an IOC
- We need to select a new analysis and fill in the IOC info

Run analysis

TLP * AMBER

PAP * AMBER

Data Type * ip

Data * 45.134.83.29

Analyzers * ☒ MISP_2_1

Cancel * Required field

Start

- We get the complete details of IOC if available

Job report

Parameters

```
{}
```

Report

```
{
  "summary": {
    "taxonomies": [
      {
        "level": "info",
        "namespace": "MISP",
        "predicate": "Search",
        "value": "0 events"
      }
    ]
  },
  "full": {
    "results": [
      {
        "url": "https://10.0.2.15/",
        "name": "misp1",
        "result": []
      }
    ]
  }
}
```

4.11 Import and Export events

- We can export the events we have created. Again we can import new events from a file to share in the community.

Import from MISP Export File

Paste MISP XML or JSON file content

or choose MISP XML or JSON file

Browse...

 misp.stix2.ORGNAME.json

☐ Publish imported events

Upload

Export

Export functionality is designed to automatically generate signatures for intrusion detection systems. To enable signature generation for a given attribute, Signature field of this attribute must be set to Yes. Note that not all attribute types are applicable for signature generation, currently we only support NDS signature generation for IP, domains, host names, user agents etc., and hash list generation for MD5/SHA1 values of file artifacts. Support for more attribute types is planned.

Simply click on any of the following buttons to download the appropriate data.

Type	Last Update	Description	Outdated	Filesize	Progress	Actions
JSON	N/A	Click this to download all events and attributes that you have access to in MISP JSON format. Attachments are disabled on this instance.	Yes	N/A	N/A	<div>Download</div> <div>Generate</div>
Snort	N/A	Click this to download all network related attributes that you have access to under the Snort rule format. Only published events and attributes marked as OS Signature are exported. Administration is able to maintain a allowlist containing host, domain name and IP numbers to exclude from the NDS export.	Yes	N/A	N/A	<div>Download</div> <div>Generate</div>
Iba	N/A	Click this to download all network related attributes that you have access to under the Iba rule format. Only published events and attributes marked as OS Signature are exported. Administration is able to maintain a allowlist containing host, domain name and IP numbers to exclude from the NDS export.	Yes	N/A	N/A	<div>Download</div> <div>Generate</div>
STIX	N/A	Click this to download a STIX document containing the STIX version of all events and attributes that you have access to. Attachments are disabled on this instance.	Yes	N/A	N/A	<div>Download</div> <div>Generate</div>
STIX2	14 minutes ago	Click this to download a STIX2 document containing the STIX2 version of all events and attributes that you have access to. Attachments are disabled on this instance.	No	13kB	Completed	<div>Download</div> <div>Generate</div>

Add From MISP Export Result

Event info	Result	Details
test Malware	1	Event with this UUID already exists. Event 1
firewall demo	Blocked by blocklist	

4.12 Taxonomy

- MISP taxonomies is a public repository of known vocabularies that can be used in threat information sharing.

Taxonomies

- previous

1

2

3

next -

last -

All

Enabled

Disabled

Enter value to search

Filter

ID ↑	Namespace	Description	Version	Enabled	Required	Highlighted	Active Tags	Actions
147	workflow	Workflow support language is a common language to support intelligence analysts to perform their analysis on data and information.	12	✖	<input type="checkbox"/>	<input type="checkbox"/>	0 / 28	<div></div>
146	vocabulaire-des-probabilites-estimates	Ce vocabulaire attribue des valeurs en pourcentage à certains énoncés de probabilité	3	✖	<input type="checkbox"/>	<input type="checkbox"/>	0 / 5	<div></div>
145	vmray	VMRay taxonomies to map VMRay Thread Identifier scores and artifacts.	1	✖	<input type="checkbox"/>	<input type="checkbox"/>	0 / 11	<div></div>
144	veris	Vocabulary for Event Recording and Incident Sharing (VERIS)	2	✖	<input type="checkbox"/>	<input type="checkbox"/>	0 / 1992	<div></div>
143	use-case-applicability	The Use Case Applicability categories reflect standard resolution categories, to clearly display alerting rule configuration problems.	1	✖	<input type="checkbox"/>	<input type="checkbox"/>	0 / 8	<div></div>
142	unified-kill-chain	The Unified Kill Chain is a refinement to the Kill Chain.	1	✖	<input type="checkbox"/>	<input type="checkbox"/>	0 / 19	<div></div>
141	type	Taxonomy to describe different types of intelligence gathering discipline which can be described the origin of intelligence.	1	✖	<input type="checkbox"/>	<input type="checkbox"/>	0 / 11	<div></div>
140	trust	The Indicator of Trust provides insight about data on what can be trusted and known as a good actor. Similar to a whitelist but on steroids, reusing features one would use with Indicators of Compromise, but to filter out what is known to be good.	1	✖	<input type="checkbox"/>	<input type="checkbox"/>	0 / 12	<div></div>
139	tor	Taxonomy to describe Tor network infrastructure	1	✖	<input type="checkbox"/>	<input type="checkbox"/>	0 / 4	<div></div>
138	tlp	The Traffic Light Protocol (TLP) (v2.0) was created to facilitate greater sharing of potentially sensitive information and more effective collaboration. Information sharing happens from an	9	✖	<input type="checkbox"/>	<input type="checkbox"/>	0 / 8	<div></div>

4.13 Galaxy

- Galaxies in MISP are a method used to express a large object called a cluster that can be attached to MISP events or attributes.

Galaxy index























← previous

1

2

next →

last →

All Enabled Disabled						Enter value to search			Filter
ID ↑	Icon	Name	Version	Namespace	Description	Enabled	Local Only	Actions	
64		UAVs/UCAVs	1	misp	Unmanned Aerial Vehicles / Unmanned Combat Aerial Vehicles	✓	✗		
63		Tool	3	misp	Threat actors tools is an enumeration of tools used by adversaries. The list includes malware but also common software regularly used by the adversaries.	✓	✗		
62		Threat Actor	3	misp	Threat actors are characteristics of malicious actors (or adversaries) representing a cyber attack threat including presumed intent and historically observed behaviour.	✓	✗		
61		Tea Matrix	1	tea-matrix	Tea Matrix	✓	✗		
60		TDS	4	misp	TDS is a list of Traffic Direction System used by adversaries	✓	✗		
59		Target Information	1	misp	Description of targets of threat actors.	✓	✗		
58		Surveillance Vendor	1	misp	List of vendors selling surveillance technologies including malware, interception devices or computer exploitation services	✓	✗		
57		Stealer	1	misp	Malware stealer galaxy.	✓	✗		
56		SoD Matrix	1	sod-matrix	SoD Matrix	✓	✗		
55		Dark Patterns	1	misp	Social Engineering - Dark Patterns	✓	✗		
54		Sigma-Rules	1	misp	Sigma Rules are used to detect suspicious behaviors related to threat actors, malware and tools	✓	✗		

5 unlisted youtube video link created by us

- <https://youtu.be/fQAXdNxt4l8>

6 Resources

- https://hdoc.csirt-tooling.org/tq-qyvTQTLeZ0wy-OPXjiw?view&fbclid=IwAR2ezxXYHF-PLIJ33tyQ2S_CwnT1dpoL_2GOGEuwRUBVNDcuIcBLk3A8J34
- Cortex installation guide: <https://github.com/TheHive-Project/CortexDocs/blob/master/installation/install-guide.md>