MISP - User Guide
A Threat Sharing Platform

A collaborative effort from the MISP community
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1.1</td>
</tr>
<tr>
<td>Book Convention</td>
<td>1.2</td>
</tr>
<tr>
<td>Quick Start</td>
<td>1.3</td>
</tr>
<tr>
<td>Requirements</td>
<td>1.4</td>
</tr>
<tr>
<td>Get Your Instance</td>
<td>1.5</td>
</tr>
<tr>
<td>General Layout</td>
<td>1.6</td>
</tr>
<tr>
<td>General Concepts</td>
<td>1.7</td>
</tr>
<tr>
<td>User Management and Global Actions</td>
<td>1.8</td>
</tr>
<tr>
<td>Using the System</td>
<td>1.9</td>
</tr>
<tr>
<td>Delegation of Event</td>
<td>1.10</td>
</tr>
<tr>
<td>Extending Events</td>
<td>1.11</td>
</tr>
<tr>
<td>Administration</td>
<td>1.12</td>
</tr>
<tr>
<td>Managing Feeds</td>
<td>1.13</td>
</tr>
<tr>
<td>Updating Python dependencies</td>
<td>1.14</td>
</tr>
<tr>
<td>Automation and MISP API</td>
<td>1.15</td>
</tr>
<tr>
<td>PyMISP - Python Library to Access MISP</td>
<td>1.16</td>
</tr>
<tr>
<td>Create an Event Based on a Report</td>
<td>1.17</td>
</tr>
<tr>
<td>Taxonomies</td>
<td>1.18</td>
</tr>
<tr>
<td>Galaxies</td>
<td>1.19</td>
</tr>
<tr>
<td>Sightings</td>
<td>1.20</td>
</tr>
<tr>
<td>Warning lists</td>
<td>1.21</td>
</tr>
<tr>
<td>Notice lists</td>
<td>1.22</td>
</tr>
<tr>
<td>Categories and Types</td>
<td>1.23</td>
</tr>
<tr>
<td>Synchronisation/Sharing</td>
<td>1.24</td>
</tr>
<tr>
<td>External Connectors</td>
<td>1.25</td>
</tr>
<tr>
<td>Modules</td>
<td>1.26</td>
</tr>
<tr>
<td>ZeroMQ - MISP publish-subscribe</td>
<td>1.27</td>
</tr>
<tr>
<td>Translations - i18n &amp; l10n</td>
<td>1.28</td>
</tr>
<tr>
<td>FAQ</td>
<td>1.29</td>
</tr>
<tr>
<td>Dev FAQ</td>
<td>1.30</td>
</tr>
<tr>
<td>Appendices</td>
<td>1.31</td>
</tr>
</tbody>
</table>
Introduction
User guide for MISP (Malware Information Sharing Platform) - An Open Source Threat Intelligence Sharing Platform. This user guide is intended for ICT professionals such as security analysts, security incident handlers, or malware reverse engineers who share threat indicators using MISP or integrate MISP into other security monitoring tools. The user guide includes day-to-day usage of the MISP's graphical user interface along with its automated interfaces (API), in order to integrate MISP within a security environment.

Acknowledgement

The MISP user guide is a collaborative effort between all the contributors to MISP including:

- Belgian Ministry of Defence (CERT)
- CIRCL Computer Incident Response Center Luxembourg
- Iklody IT Solutions
- NATO NCIRC
- Cthulhu Solutions
- CERT-EU

and many other contributors especially the ones during the MISP hackathons.

Contributing

We welcome contributions to the MISP book. If you want to contribute, fork the misp-book repository and pull a request with your changes. You can also open issues if you find any errors or propose changes.
Format

MISP book is available in HTML, PDF, ePub and Kindle mobi format.

License

The MISP user guide is dual-licensed under GNU Affero General Public License version 3 and CC-BY-SA 4.0 international.

- Copyright (C) 2012 Christophe Vandeplas
- Copyright (C) 2012 Belgian Defence
- Copyright (C) 2012 NATO / NCIRC
- Copyright (C) 2013-2020 Andras Iklody
- Copyright (C) 2015-2020 Alexandre Dulaunoy
- Copyright (C) 2014-2020 CIRCL - Computer Incident Response Center Luxembourg
- Copyright (C) 2018 Camille Schneider
- Copyright (C) 2018-2020 Steve Clement
**description: Convention Used in MISP-Book**

**Convention Used in This Book**

A **code block or value** is used for variable, function or menu names in MISP.

**Language**

The language in this book is American English. All the screenshots and examples are in English.

**CoC**

The same code of conduct applies to this book as for the main MISP project. As a book can sometimes be considered the inadvertent soul of a piece of software, please take good care and consideration of our [Code of Conduct](#). The CoC can be read [here](#).

**Example install**

The examples and screenshots provided in this book have been created with the MISP Autogenerated VM.

To get a copy of the latest VM [click here](#).

**MISP Instance**

In general when talking about a network of inter-connected MISP servers, each server is a MISP instance. Whilst we have no strong feelings towards anyone's naming schemes, as a rule of thumb try to have a scheme that makes everyday use easy when analysts need to talk about remote MISP instances.

The hostname used for the instance in this book is `misp.local` and we will henceforth refer to it either by name or as local MISP instance.

**Example Organisations**

As MISP is a platform to support information sharing, example organisations are often used within this book.

A set of users and organisations are used in the different examples.

The following two organisations are regularly used as example:

- **Setec Astronomy with UUID** 5d838399-7b24-4386-bb4d-4c8f958d210f
- **Acme Finance with UUID** 5d83823e-eda8-443a-9fa8-4e12956d210f
Starting from MISP 2.4.71, the example organisations with the above mentioned UUID are black-listed to avoid large distribution of sample events while testing a MISP instance. If you want to test your distribution, the sample organisation black-listing can be removed in Administration / Manage Org Blacklists.

Example IOCs

As with the example organisations, we want to make this book as useful as possible by using real life examples.

The following IOC examples have been used:

- Sirefef (aka ZeroAccess) Sample Event ID: #31337
- WannaCry Sample Event ID: #42
- Dridex Sample Event ID: #23
Quick Start
MISP (Open Source Threat Intelligence and Sharing Platform) software facilitates the exchange and sharing of threat intelligence, Indicators of Compromise (IoCs) about targeted malware and attacks, financial fraud or any intelligence within your community of trusted members. MISP sharing is a distributed model containing technical and non-technical information which can be shared within closed, semi-private or open communities. Exchanging such information should result in faster detection of targeted attacks and improve the detection ratio, whilst also reducing the number of false positives.

With the focus on automation and standards, MISP provides you with a powerful ReST API, extensibility (via misp-modules) or additional libraries such as PyMISP, jump ahead to these chapters to get started.

**Login into MISP**

MISP default credentials:

<table>
<thead>
<tr>
<th>Username:</th>
<th><a href="mailto:admin@admin.test">admin@admin.test</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Password:</td>
<td>admin</td>
</tr>
</tbody>
</table>

**I forgot my admin password :(**

You can quickly reset it via the command line. You need to know the Admins E-Mail address. Replace www-data with whoever runs the webserver.

```bash
sudo -u www-data /var/www/MISP/app/Console/cake Password admin@admin.test Password1234
```

**Tasks to do after first Start**

1. Change site admin password
2. Activate Feeds
3. Setup your User 3.1 Designate a Site Admin and an Org Admin 3.2 Add some contributing users and assign the corresponding Roles
4. MISP Administration 4.1 Edit your first organisations’ name

**Password Policy**

- [12]: Ensure that the password is at least 12 characters long
- [A-Z]: contains at least one upper-case
- [0-9]: includes a digit or a special character
- [a-z]: at least one lower-case character.

If you need a password generator use:

- Ubuntu / Debian: `pwgen`
- Website: [LastPass PW Generator](#)
- Built-in generator in Keepass* and other password manager
- Built-in generator in various web browsers

**All Generator tools are only possibilities without any guarantee!**
tl;dr

Create an Event

A. Add Event

1. List Events
2. Populate Fields

B. Add Attachments

3. GIPE sandbox
4. Add Attachment

C. Add Event Attributes

5. Populate Fields

The following attribute types should be added for each event:
- ip-src: source IP of attacker
- email-src: email used to send malware
- md5/sha1/sha256: checksum
- Hostname: full host/dnsname of attacker
- Domain: domain name used in malware

6. Add Attribute

7. Add Attachment

8. Populate Fields

9. Upload
Browse Past Events

1. List Events
2. Filter
3. Click any row
4. See events with one or more matching attributes
Export Events for logsearches

1. Return

2. Download for log correlation
Create an Event

1. The event created will be restricted to the organisations included in the distribution setting on the local instance only until it is published.

Add Event

Date
2018-05-10

Distribution
This community only

Threat Level
High

Analysis
Initial

Event Info
Quick Event Description or Tracking Info

Extends event
Event UUID or ID. Leave blank if not applicable.

GFI sandbox
Choose file
No file chosen

Add

2. Summarized description:
- Distribution
- Threat Level
- Event Info
- GFI sandbox (optional)
- Does it extend? (optional)

3. Add == Save
You only have to add a few pieces of information to register your Event. Further details will be specified after the Event has been added.

**Describe Event**
Now you can specify the information for your Event (you will need to scroll the window).

**Free-Text Import Tool**

All IoC data entered is made up of an event object and described by its connected attributes.

To get straight to the Freetext import tool click here.
The following will pop-up.

If you have a list of indicators from which you would like to quickly generate attributes then the Free-text import tool is just what you need. Simply paste your list of indicators (separated by line-breaks) into this tool.
## Freetext 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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Similar Attributes</th>
<th>Category</th>
<th>Type</th>
<th>IDS</th>
<th>Comment</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>01e21a06a1f</td>
<td></td>
<td>Payload delivery</td>
<td>sha256</td>
<td></td>
<td></td>
<td>Imported via the Freetext Import</td>
</tr>
</tbody>
</table>

**Submit**

<table>
<thead>
<tr>
<th>sha256</th>
<th>authentihash</th>
<th>Change all</th>
</tr>
</thead>
</table>

Update all comment fields
The tool will help you to find similarities between your import and other issues already registered in MISP.

Event ID: 95
Event info: OSINT - LinkedIn information used to spread banking malware in the Netherlands
Category: Payload delivery
Type: filename|sha256
Value: office.bin|c1e21a05a1fa1de298392668b5910c
Comment: downloaded malware
For example, you can see the ID of all related Events and view their information.

Alternative to import

An alternative route to reach the Freetext import tool is shown below.
For Freetext import select it
Tags and Taglist

Using existing Data

Another easy way to add information is to use Tags. You can see the result of adding existing Tags (circl:incident-classification=XSS ans circl:incident-classification="information-leak").

To add tags from a Taxonomy or Custom tags, click here.
By clicking the button, you can add more tags from an existing Taglist.

\[\text{Select Tag collections (taxonomies) or self-created tags} \]

\[\text{Select the input box to see the tags} \}

/\ If no tags show up, enable a Taxonomy or create some custom tags
In particular the "Taxonomy Library: circl" Taglist is very complete.

Once you added the tag(s) it will show in your main event window and in the list event view.

### OSINT - Threat Spotlight: Ratsnif - New Network Vermin...

<table>
<thead>
<tr>
<th>Event ID</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>UUID</td>
<td>5d2417e3-f448-4d33-bbdd-2a1938a6ac88</td>
</tr>
<tr>
<td>Creator org</td>
<td>ORGNAME</td>
</tr>
<tr>
<td>Owner org</td>
<td>ORGNAME</td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:admin@admin.test">admin@admin.test</a></td>
</tr>
<tr>
<td>Tags</td>
<td><img src="malware" alt="malware" /></td>
</tr>
<tr>
<td>Date</td>
<td>2019-07-09</td>
</tr>
<tr>
<td>Threat Level</td>
<td>Undefined</td>
</tr>
<tr>
<td>Analysis</td>
<td>Initial</td>
</tr>
</tbody>
</table>

Once you have confirmed the tag(s) they will appear here
Local tags

Local tags can be added in a similar fashion.
They will be identified by a corresponding icon.

<table>
<thead>
<tr>
<th>Tag</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>type:OSINT</td>
<td>x</td>
</tr>
<tr>
<td>osint:lifetime=&quot;perpetual&quot;</td>
<td>x</td>
</tr>
<tr>
<td>osint:source-type=&quot;blog-post&quot;</td>
<td>x</td>
</tr>
<tr>
<td>osint:certainty=&quot;93&quot;</td>
<td>x</td>
</tr>
<tr>
<td>estimative-language:confidence-in-analytic-judgment=&quot;high&quot;</td>
<td>x</td>
</tr>
<tr>
<td>workflow:todo=&quot;review-for-privacy&quot;</td>
<td>x</td>
</tr>
</tbody>
</table>

**Date** | 2019-07-04

**Threat Level** | Low

**Analysis** | Ongoing

**Distribution** | All communities
No tags in list

In case you get the below. You need to either enable an existing Taxonomy or add some custom tags.

//\ If no tags show up, enable a Taxonomy or create some custom tags
Make your own Taglist

If you want to make your own Taglist, select Add Tag.
You will see the following window:
Then, when you add the new tag it will appear in the Custom Taglist.

**Suggestions**

The following attribute types should be added for each Event:

- `ip-src`: source IP of attacker
- `email-src`: email used to send malware
- `md5/sha1/sha256`: checksum
- `Hostname`: full host/dnsname of attacker
- `Domain`: domain name used in malware

**Browsing Events**

To see your Event, select List Events from the menu Events Action. You can click any row and select a filter.
If you click on your Event's number, you can see all the information related to your Event.

<table>
<thead>
<tr>
<th>Event ID</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>UUID</td>
<td>5d2417e3-448-4d3-b0dd-2a19386ed68f</td>
</tr>
<tr>
<td>Creator org</td>
<td>ORGNAME</td>
</tr>
<tr>
<td>Owner org</td>
<td>ORGNAME</td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:admin@admin.test">admin@admin.test</a></td>
</tr>
<tr>
<td>Tags</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>2019-07-09</td>
</tr>
<tr>
<td>Threat Level</td>
<td>Undefined</td>
</tr>
<tr>
<td>Analysis</td>
<td>Initial</td>
</tr>
</tbody>
</table>

Related Events
- Unidentified Malware via SpamMailServer
  - Date: 2019-07-09

This is the Organization's name

Related events, events that share attributes, will be displayed here.
Export Events for Log Search

Export functionality is designed to automatically generate signatures for intrusion detection systems. To enable signature generation for a given attribute, the 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 NIDS 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 for log correlation.

### Export

Export functionality is designed to automatically generate signatures for intrusion detection systems. To enable signature generation for a given attribute, Signature field of the attribute must be set to 'Yes'. Note that not all attribute types are applicable for signature generation; currently we only support 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.

<table>
<thead>
<tr>
<th>Type</th>
<th>Last Update</th>
<th>Description</th>
<th>Outlined</th>
<th>Progress</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML</td>
<td>N/A</td>
<td>Click this to download all events and attributes that you have access to (except file attachments) in a custom XML format.</td>
<td>Yes</td>
<td>N/A</td>
<td>Download Generate</td>
</tr>
<tr>
<td>CSV_Sig</td>
<td>N/A</td>
<td>Click this to download all attributes that are indicators and that you have access to (except file attachments) in CSV format.</td>
<td>Yes</td>
<td>N/A</td>
<td>Download Generate</td>
</tr>
<tr>
<td>CSV</td>
<td>N/A</td>
<td>Click this to download all attributes that are indicators and that you have access to (except file attachments) in CSV format.</td>
<td>Yes</td>
<td>N/A</td>
<td>Download Generate</td>
</tr>
</tbody>
</table>
Enable a Taxonomy

Enable and fetch a feeds
MISP Instance requirements

- Intro
  - The biggie
    - Tool assisted sizing

Intro

There are various ways you can run a MISP instance.

- Virtualized with docker/ansible/packer etc
- VMware/Virtualbox/Xen etc
- Dedicated hardware
- Road warrior setups
- Air-gapped setups

Whilst there is never an ultimate answer to what specifications a system needs, we try to give an approximate answer depending on your use case.

The biggie

Having millions of events with millions of attributes (indicators) will eventually result in sub-par performance. Ideally you have millions of attributes and thousands of events. But this also depends on how you ingest the data. With millions of attributes a bottleneck could be the correlation engine. Especially if you have many duplicates in your events. (Use the feed matrix to see if feeds are massively overlapping)

Tool assisted sizing

During a hackathon misp-sizer was conceived. (code) This can give you a very rough estimate and needs some more improvements.
Get your own MISP instance

The intention of this chapter is to support you in getting your own MISP instance up and running.

**MISP Virtual Machine**

CIRCL maintains the image of a recent MISP virtual machine online. This VM is generated after every commit to the main MISP repository on Github.

This is a very easy out of the box solution, optimized for product evaluation and to support trainings held by CIRCL staff.

**MISP VM Download**

The best place to get the latest version of the MISP virtual machine, as well as all the available training materials is the MISP training materials page on the CIRCL website.

If you do not remember the direct link to the MISP training materials here are the very easy to remember steps you have to follow to reach the right place:

1. Access the CIRCL homepage
2. Navigate to the Training area
3. Click MISP Malware Information Sharing Platform - Threat Sharing
4. Follow the link to the Training materials freely available

Download the image of the virtual machine and validate the SHA512 fingerprint.

**Import Appliance**

In VirtualBox use the "Import Appliance..." functionality to import the virtual machine.
VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below.

MISP_v2.4.70.ova
The instructions in this manual covers VirtualBox only. If you prefer another virtualization solution like VMWare you can find some quick instruction on the MISP training materials page.

ESXi Servers have been tested too. Should work without problem but some manual changing of the ATA-Bus is needed.

**MISP VM Credentials**

The MISP image is pre-configured to be reachable on the private IP address **localhost** by SSH on port 2222. The GUI is reachable by http://localhost:8080/.

You should have two interfaces on your VirtualBox configuration (NAT and host-only). You can also configure access to the MISP instance by doing port forwarding on the NAT interface.

MISP credentials:

- **GUI Admin:** admin@admin.test:admin (it's the site admin account with full rights, feel free to create other users)
- **Shell/SSH:** misp : Password1234
- **MySQL:** The credentials are generated during the VM generator. The details are located in ~misp/mysql.txt

**Networking on the VM**

Virtualbox has a neat feature to forward ports from your Host machine to the Guest VM. We forward the following ports:

- ssh Forward from 2222 on Host -> 22 on guest
- http Main WebUI - 8080 on Host -> 80 on guest
- https Not in use - 8443 on Host -> 443 on guest
- 8001 MISP Dashboard - 8001 on Host -> 8001 on guest
- 8888 Viper Web UI - 8888 on Host -> 8888 on guest
- 1666 misp-modules used to poll the misp-modules API - 1666 on Host -> 6666 on guest

If the port is already used on your host, virtualbox will still boot and all the other ports will work.

To change the port forwarding select the running VM in the UI and click on **Settings -> Network -> Advanced -> Port forwarding**.
Overview of default port forwards

<table>
<thead>
<tr>
<th>Name</th>
<th>Protocol</th>
<th>Host IP</th>
<th>Host Port</th>
<th>Guest IP</th>
<th>Guest Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>dashboard</td>
<td>TCP</td>
<td>8001</td>
<td>0.0.0.0</td>
<td>8001</td>
<td></td>
</tr>
<tr>
<td>http</td>
<td>TCP</td>
<td>8080</td>
<td>80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>https</td>
<td>TCP</td>
<td>8443</td>
<td>443</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misp-modules</td>
<td>TCP</td>
<td>1666</td>
<td>0.0.0.0</td>
<td>6666</td>
<td></td>
</tr>
<tr>
<td>ssh</td>
<td>TCP</td>
<td>2222</td>
<td>0.0.0.0</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>viper</td>
<td>TCP</td>
<td>8888</td>
<td>0.0.0.0</td>
<td>8888</td>
<td></td>
</tr>
</tbody>
</table>
The reason that some entries have 0.0.0.0 and other are left blank is due to a virtualbox bug where traffic would not be sent to the Guest VM.
VMWare users will need to connect to whatever IP the VM has on your host. There is NO port forwarding done for VMWare.

**Potential issues**

You might have a very old VM installed and the ports are not be forwarded. Either configure the port forwards manually or download a new VM.
General Layout

The top bar

Simple User
This menu contains all of the main functions of the site as a series of dropdown menus. These contains all (from the current simple user's perspective) accessible functions sorted into several groups.

- **Home button**: This button will return you to the start screen of the application, which is the event index page (more about this later).
- **Event Actions**: All the malware data entered into MISP is made up of an event object that is described by its connected attributes. The Event actions menu gives access to all the functionality that has to do with the creation, modification, deletion, publishing, searching and listing of events and attributes.
- **Galaxies**: Shortcut to the list of MISP Galaxies on the MISP instance.
- **Input Filters**: Input filters alter what and how data can be entered into this instance. Apart from the basic validation of attribute entry by type, it is possible for the site administrators to define regular expression replacements and blacklists for certain values in addition to blocking certain values from being exportable. Users can view these replacement and blacklist rules here whilst administrator can alter them.
- **Global Actions**: This menu gives you access to information about MISP and this instance. You can view and edit your own profile, view the manual, read the news or the terms of use again, see a list of the active organizations on this instance and a histogram of their contributions by attribute type.
- **MISP**: Simple link to your BASEURL
- **Steve**: Name (Auto generated from Mail address) of current logged in user
- **Envelope**: Link to User Dashboard where you can consult some of your notifications and changes since last visit. Like some of the proposals received for your organisation.
- **Log out**: The Log out button to end your session immediatly.

**Admin Menu Bar**

<table>
<thead>
<tr>
<th>Home</th>
<th>Event Actions</th>
<th>Galaxies</th>
<th>Input Filters</th>
<th>Global Actions</th>
<th>Sync Actions</th>
<th>Administration</th>
<th>Audit</th>
<th>MISP</th>
<th>Admin</th>
<th>Log out</th>
</tr>
</thead>
</table>

---

43
- **Home button**: idem as user.
- **Event Actions**: ibidem
- **Galaxies**: You can additionally update the Galaxies.
- **Input Filters**: Ibidem
- **Global Actions**: Ibidem
- **Sync Actions**: With administrator access rights, shows a list of the connected instances and allows the initiation of a push and a pull (more about the synchronization mechanisms later).
- **Administration**: Administrators can add, edit or remove user accounts and user roles. Roles define the access rights to certain features such as publishing of events, usage of the REST interface or synchronization of any user belonging to the given role. Site administrators can also access a contact form, through which it is possible to reset the passwords of users, or to just get in touch with them via encrypted e-mails.
- **Audit**: If you have audit permissions, you can view the logs for your organization (or for site admins for the entire system) here or even search the logs if you are interested in something specific.
- **MISP**: Ibidem
- **Admin**: Ibidem
- **Envelope**: Link to User Dashboard where you can consult some of your notifications and changes since last visit. Like some of the proposals received for your organisation.
- **Log out**: The Log out button to end your session immediately.

**A list of the contents of each of the above drop-down menus**

**Event actions**
Event Actions

List Events
Add Event
List Attributes
Search Attributes
REST client

View Proposals
Events with proposals

List Tags
Add Tag
List Taxonomies

List Templates
Add Template

Export
Automation
- **List Events**: Lists all the events in the system that are not private or belong to your organisation. You can add, modify, delete, publish or view individual events from this view.

- **Add Event**: Allows you to fill out an event creation form and create the event object, which you can start adding attributes.

- **List Attributes**: Lists all the attributes in the system that are not private or belong to your organisation. You can modify, delete or view each individual attribute from this view.

- **Search Attributes**: You can set search terms for a filtered attribute index view here.

- **REST client MISP Online REST client** where you can make calls directly to the AI via a Web UI.

- **View Proposals**: Shows a list of all proposals that you are eligible to see.

- **Events with proposals**: Shows all of the events created by your organisation that has pending proposals.

- **List Tags**: List all the tags that have been created by users with tag creation rights on this instance.

- **Add Tag**: Create a new tag.

- **List Taxonomies**: List all of the taxonomies installed on the **MISP instance**. This is also the place to activate the taxonomies as a **Org Admin/Site Admin**.

- **List Templates**: List all of the templates created by users with template creation rights on this instance.

- **Add Template**: Create a new template.

- **Export**: Export the data accessible to you in various formats.

- **Automation**: If you have authentication key access, you can view how to use your key to use the REST interface for automation here.

**Input filters**

<table>
<thead>
<tr>
<th>Input Filters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Import Regexp</strong></td>
</tr>
<tr>
<td><strong>Signature Whitelist</strong></td>
</tr>
<tr>
<td><strong>List Warninglists</strong></td>
</tr>
<tr>
<td><strong>List Noticelists</strong></td>
</tr>
</tbody>
</table>
- **Import Regexp**: You can view the Regular Expression rules, which modify the data that can be entered into the system. This can and should be used to help filter out personal information from automatic imports (such as removing the username from windows file paths), having unified representation for certain common values for easier correlation or simply standardizing certain input. It is also possible to block certain values from being inserted. As a site administrator or a user with regex permission, you can also edit these rules.

- **Signature Whitelist**: You can view the whitelist rules, which contains the values that are blocked from being used for exports and automation on this instance. Site administrators have access to editing this list.

- **List Warninglists**: MISP warninglists are lists of well-known indicators that can be associated to potential false positives, errors or mistakes. The warning lists are integrated in MISP to display an info/warning box at the event and attribute level.

- **List Noticelists**: MISP noticelists are lists of #Todo: Double check description from repo!!!
**News:** Read about the latest news regarding the MISP system

**My Profile:** Manage your user account.

**Dashboard:** allow you to see your Notifications of Proposals, Events with proposals and Delegation request. Your can see the last changes since your last visit, as Events updates and Events publications.

**Organizations:** View the organizations having a presence on this instance, with some useful informations as contact's name.

**Role Permissions:** You can view the role permissions here.

**List Sharing Groups:** You can view the list of existing Sharing Groups who you or your organization have access.

**Add Sharing Group:** You can create a sharing group.

**User Guide:** A link to this user guide.

**Categories & Types:** Quick overview of Attribute Categories and Type. e.g: md5 -> Payload delivery, Artifacts dropped, Payload installation, External analysis

**Terms & Conditions:** General terms and conditions which can be configured in Administration -> Server Settings -> MISP Settings: MISP.terms_file. From the UI: "The filename of the terms and conditions file. Make sure that the file is located in your MISP/app/files/terms directory"

**Statistics:** View a series of statistics about the users and the data on this instance.

**List Discussions:** List threads of discussions created on the MISP instance by the organisations connected to this local community.

**Start Discussion:** Create a new discussion thread.

### Sync Actions

- **List Servers**
- **List Feeds**
- **List Servers**: Connect your MISP instance to other instances, or view and modify the currently established connections.
- **List Feeds**: Follow the RSS feeds of other organization or CERTs worldwide.

**Administration**

- List Users
- Add User
- Contact Users
- List Organisations
- Add Organisation
- List Roles
- Add Role
- Server Settings & Maintenance
- Jobs
- Scheduled Tasks
- Blacklist Event
- Manage Event Blacklists
- Blacklist Organisation
- Manage Org Blacklists
- **List Users**: View, modify or delete the currently registered users.
- **Add User**: Create an account for a new user for your organisation. Site administrators can create users for any organisation.
- **Contact Users**: You can use this view to send messages to your current or future users or send them a temporary password. When adding a new user to the system, or when you want to manually reset the password for a user, just use the "Send temporary password" setting.

  After selecting the action, choose who the target of the e-mails should be (all users, a single user or a user not yet in the system).

  You can then specify (if eligible) what the e-mail address of the target is (for existing users you can choose from a dropdown menu).

  In the case of a new user, you can specify the future user's GnuPG key, to send his/her new key in an encrypted e-mail.

  The system will automatically generate a message for you, but it is also possible to write a custom message if you tick the checkbox, but don't worry about assigning a temporary password manually, the system will do that for you, right after your custom message.

- **List Organisations**: View the organizations having a presence on this instance, with some useful informations.
- **Add Organisation**:  
  - **List Roles**: List, modify or delete currently existing roles.
- **Add Role**: Create a new role group for the users of this instance, controlling their privileges to create, modify, delete and to publish events and to access certain features such as the logs or automation.
- **Server Settings & Maintenance**: Various tools, upgrade scripts that can help a site-admin run the instance & Set up and diagnose your MISP installation.
- **Jobs**: View the background jobs and their progress
- **Scheduled Tasks**: Schedule the pre-defined tasks for your instance (this currently includes export caching, server pull and server push).
- **Blacklist Event**: Link to form where you can quickly add an event to a blacklist with it's UUID.
- **Manage Event Blacklists**: List of blacklisted events on MISP instance.
- **Blacklists Organisation**: Link to for where you can quickly add an organisation to a blacklist with it's UUID.
- **Manage Org Blacklists**: List of blacklisted Organisations on this instance.

**Audit**

![Audit Menu]

- **List Logs**
- **Search Logs**
- **List Logs**: View the logs of the instance.
- **Search Logs**: Search the logs by various attributes.

**The left bar**

This bar changes based on each page-group. The blue selection shows you what page you are on.
General Concepts

Admins and Site Admins

There are two types of admins in MISP: Admins (also referred to as org admins) and Site Admins. Whilst the former can only do some limited administration of users of his/her own organisation, site admins have access to all of the features and data of the system. They are in charge of making sure that the system runs correctly and the maintenance of MISP.

Background Jobs

A lot of the heavier tasks are a burden to users, in that their actions can cause long delays (and in some cases timeouts) while the application logic is executing. To alleviate this, long processes have been (if enabled) moved to background jobs, meaning that their execution happens asynchronously in the background, allowing the user to freely interact with the platform whilst the request is being processed.

MISP Instance

A MISP instance is an installation of the MISP software and the connected database. All the data visible to the users is stored locally in the database and data that is shareable (based on the distribution settings) can be synchronised with other instances via the Sync actions. The instance that you are reading this manual on will be referred to as "this instance" or "your instance". The instances that your instance synchronises with will be referred to as "remote instances".

Organisation administrators and Site administrators

We have two types of administrators, site and organisation admins. The former has access to every administrator feature for all the data located on the system including global features such as the creation and modification of user roles and instance links, whilst organisation admins can administer users, events and logs of their own respective organisations.

Pivot path

The (branching) path taken by a user from event to event while following correlation links. This is represented by the branching graph in the event view.
Pivoting

The act of navigating from event to event through correlation links.

Proposals

Each event can only be directly edited by users of the original creator organisation (and site admins). However, if another organisation would like to amend an event with extra information on an event, or if they’d like to correct a mistake in an attribute, they can create a Proposal. These proposals could then be accepted by the original creator organisation. These proposals can be pulled to another server, allowing users on connected instances to propose changes which then could be accepted by the original creators on another instance (and subsequently pushed back).

Publishing

When an event is first created by a user, it is visible to everyone on the instance based on the access rights ("Your organisation only" events will not be visible to users of other organisations), but they will not be synchronised and they won’t be exportable. For this, a user with publishing permission of the organisation that created the event has to publish the event. The system will then inform all the users of the instance that are subscribing to e-mail notifications and who have access to view the published event via an e-mail.

Pull

Pulling is the process of using the configured sync user on a remote instance to REST GET all of the accessible data (based on the distribution rights) to your instance and store it.

Push

Pushing is the process of using a configured instance link to send an event or all accessible events (limited by the distribution rights) through the REST interface to a remote instance.

Scheduled Tasks

Certain common tasks can be scheduled for a later execution or for regular recurring executions. These tasks currently include caching all of the export formats, pulling from all eligible instances and pushing to all eligible instances.

Sync User

A user of a role that grants sync permissions, these users (and their authentication keys) are used to serve as the points of connection between instances. Events pushed to an instance are pushed to a sync user, who then creates the events on the remote instance. Events pulled are added by the sync user that is used to connect the remote instance to your instance. As an administrator, keep in mind that a sync user needs auth key and publish permissions, has to have undergone the mandatory password change and has to have accepted the Terms of Use in order for the sync to work. Please make sure that all of these steps are taken before attempting to push or pull.

Synchronisation

What we call synchronisation is an exchange of data between two (or more) MISP instances through our pull and push mechanisms.

Tagging
Users with tagging rights can assign various dynamically created tags to events, allowing an arbitrary link between events to be created. It is possible to filter events based on these tags and they can also be used to filter events for the automation.

**Templating**

Users with templating rights can create easy to fill forms that help with the event creation process.
User Management and Global Actions

First run of the system

When first logging into MISP with the username and password provided by your administrator, there are a number of things that need to be done, before you can start using the system.

- **Accepting the Terms of use**: The terms of use are shown immediately after logging in for the first time, make sure to read through this page before clicking "Accept Terms" at the bottom of the page.
- **Changing the password**: After accepting the ToU, you'll be prompted to change your password, but keep in mind that it has to be pass to the MISP password policy. Enter the same password into the confirm password field, before clicking submit to finalise the change.

### Change Password

<table>
<thead>
<tr>
<th>Password</th>
<th>Confirm Password</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Submit button]
Setting up the GnuPG Key: In order for the system to be able to encrypt the messages that you send through it, it needs to know your GnuPG key. Navigate to the Edit profile view (My Profile on the left -> Edit profile in the top right corner). Paste the key into the GnuPG Key field and click submit.

Subscribing to Auto-alerts: Turning auto-alerts on will allow the system to send you e-mail notifications about any new public events entered into the system by other users and private events added by members of your organisation. To turn this on, navigate to the Edit profile view (My profile on the left navigation menu -> Edit profile in the top right corner). Tick the auto-alert checkbox and click submit to enable this feature.
• **Subscribing to e-mails sent via the “Contact Reporter” functionality:** This feature is turned on right below the autoalerts and will allow you to receive e-mails addressed to your organisation whenever a user tries to ask about an event that was posted by a user of your organisation. Keep in mind that you can still be addressed by such a request even when this setting is turned off, if someone tries to contact you as the event creator directly or your organisation for an event that you personally have created then you will be notified.

• **Reviewing the Terms & Conditions:** To review the Terms & Conditions or to read the User Guide, use the appropriate button on the left navigation menu.

• **Making sure that compatibility mode is turned off (IE9&IE10):** Compatibility mode can cause some elements to appear differently than intended or not appear at all. Make sure you have this option turned off.
Using the system

Creating an event

The process of entering an event can be split into 3 phases, the creation of the event itself, populating it with attributes and attachments and finally publishing it.

During this first step, you will create a basic event without any actual attributes, but storing general information such as a description, time and risk level of the incident. To start creating the event, click on the New Event button on the left and fill out the form you are presented with. The following fields need to be filled out:
Add Event

<table>
<thead>
<tr>
<th>Date</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All communities</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Threat Level</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>initial</td>
</tr>
</tbody>
</table>

Event Description
Quick Event Description or Tracking Info

GFI sandbox
Choose File
No file chosen

Add
- **Date:** The date when the incident has happened. Just click this field and a date-picker will pop up where you can select the desired date.

- **Distribution:** This setting controls, who will be able to see this event once it becomes published and eventually when it becomes pulled. Apart from being able to set which users on this server are allowed to see the event, this also controls whether the event will be synchronised to other servers or not. The distribution is inherited by attributes: the most restrictive setting wins. The following options are available:
  - **Your organization only:** This setting will only allow members of your organisation to see this. It can be pulled to another instance by one of your organisation members where only your organisation will be able to see it. Events with this setting will not be synchronised. Upon push: do not push. Upon pull: pull.
  - **This Community-only:** Users that are part of your MISP community will be able to see the event. This includes your own organisation, organisations on this MISP server and organisations running MISP servers that synchronise with this server. Any other organisations connected to such linked servers will be restricted from seeing the event. Upon push: do not push, Upon pull: pull and downgrade to Your organization only.
  - **Connected communities:** Users that are part of your MISP community will be able to see the event. This includes all organisations on this MISP server, all organisations on MISP servers synchronising with this server and the hosting organisations of servers that connect to those afore mentioned servers (so basically any server that is 2 hops away from this one). Any other organisations connected to linked servers that are 2 hops away from this own will be restricted from seeing the event. Upon push: downgrade to This Community only and push. Upon pull: pull and downgrade to This Community only.
  - **All communities:** This will share the event with all MISP communities, allowing the event to be freely propagated from one server to the next. Upon push: push. Upon pull: pull.
  - **Sharing group:** This will share the event to the defined sharing group. This includes only the organisations defined in the sharing group. The distribution can be local and cross-instance depending of the sharing group definition. For more information on sharing groups, refer to the sharing group section.

- **Threat Level:** This field indicates the risk level of the event. Incidents can be categorised into three different threat categories (low, medium, high). This field can alternatively be left as undefined. The 3 options are:
  - **Low:** General mass malware.
  - **Medium:** Advanced Persistent Threats (APT)
  - **High:** Sophisticated APTs and 0day attacks.

- **Analysis:** Indicates the current stage of the analysis for the event, with the following possible options:
  - **Initial:** The analysis is just beginning
  - **Ongoing:** The analysis is in progress
  - **Completed:** The analysis is complete

- **Event Description:** The info field, where the malware/incident can get a brief description starting with the internal reference. This field should be as brief and concise as possible, the more detailed description happens through attributes in the next stage of the event's creation. Keep in mind that the system will automatically replace detected text strings that match a regular expression entry set up by your server's administrator(s).

- **GFI Sandbox:** It is possible to upload the exported .zip file from GFI sandbox with the help of this tool. These will be dissected by the MISP and a list of attributes and attachments will automatically be generated from the .zip file. Whilst this does most of the work needed to be done in the second step of the event's creation, it is important to manually look over all the data that is being entered.

### Add attributes to the event

The second step of creating an event is to populate it with attributes and attachments. This can be done by adding them manually or importing the attributes from an external format (OpenIOC, ThreatConnect). To import from an external format or to upload an attachment use the options in the menu on the left.
Using the System

<table>
<thead>
<tr>
<th>Date</th>
<th>Category</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014-10-15</td>
<td>Network activity</td>
<td>inдол</td>
<td>R E E R2</td>
</tr>
</tbody>
</table>
Using the above shown buttons, you can populate an event using various tools that will be explained in the following section. Let's start with the Add Attribute button.

**Add Attribute**

Keep in mind that the system searches for regular expressions in the value field of all attributes when entered, replacing detected strings within it as set up by the server's administrator (for example to enforce standardised capitalisation in paths for event correlation or to bring exact paths to a standardised format). The following fields need to be filled out:

<table>
<thead>
<tr>
<th>Category</th>
<th>Type</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network activity</td>
<td>url</td>
<td>All communities</td>
</tr>
</tbody>
</table>

Value

http://www.teamliquid.net

Contextual Comment

- for Intrusion Detection System
- Batch Import

Submit
- **Category:** This drop-down menu explains the category of the attribute, meaning what aspect of the malware this attribute is describing. This could mean the persistence mechanisms of the malware or network activity, etc. For a list of valid categories, click here

- **Type:** Whilst categories determine what aspect of an event they are describing, the Type explains by what means that aspect is being described. As an example, the source IP address of an attack, a source e-mail address or a file sent through an attachment can all describe the payload delivery of a malware. These would be the types of attributes with the category of payload deliver. For an explanation of what each of the types looks like together with the valid combinations of categories and types, click here

- **Distribution:** This drop-down list allows you to control who will be able to see this attribute. The distribution is inherited by attributes: the most restrictive setting wins. For more info, read the distribution information in the creating an event section - click here

- **Value:** The actual value of the attribute, enter data about the value based on what is valid for the chosen attribute type. For example, for an attribute of type ip-src (source IP address), 11.11.11.11 would be a valid value. For more information on types and values, click here

- **Contextual Comment:** You can add some comments to the attribute that will not be used for correlation but instead serves as purely an informational field.

- **For Intrusion Detection System:** This option allows the attribute to be used as an IDS signature when exporting the NIDS data, unless it is being overruled by the white-list. For more information about the white-list, head over to the administration section. If the IDS flag is not set, the attribute is considered as contextual information and not to be used for automatic detection.

- **Batch import:** If there are several attributes of the same type to enter (such as a list of IP addresses, it is possible to enter them all into the same value-field, separated by a line break between each line. This will allow the system to create separate lines for the each attribute.

### Add Object

Please have a look at the MISP-objects chapter

### Create and manage Sharing Groups

Sharing groups in MISP are a more granular way to create re-usable distribution lists for events/attributes that allow users to include organisations from their own instance (local organisations) as well as organisations from directly, or indirectly connected instances (external organisations). Sharing groups can be created by any user that has the sharing group editor permission. Additionally, sharing groups can be edited by any user that has the aforementioned permission in addition to being a member of the sharing group's creating organisation, or any organisation that is marked as an "extender" of the sharing group. The main use for the extend feature is delegating the rights to add users to trusted partners. For example, when sharing with a different industry sector, knowing all actors that should receive the information is often not possible, so delegating the rights to extend the event to a trusted representative of said sector would allow for someone with more insight to find and add the proper list of partners for the sharing group.
The most general use-cases for sharing groups are creating re-usable topical subgroups in MISP that share events or for ad-hoc sharing scenarios (such as several organisations involved in a specific incident wanting to work together). Generally sharing groups add a level of complexity for the users involved as well as a performance overhead on the data marked with it.

As a best-practice recommendation, using traditional distribution methods is preferred unless they cannot cover the given use-case. Also, whilst sharing groups can be assigned to both events and attributes, it is highly recommended to use the special "inherit" distribution setting on attributes whenever the attribute's sharing group would match the event's.

Sharing groups consist of the following elements, each of which has its own page in the sharing group creator/editor tool (accessed via the Global actions -> List Sharing Groups and Add Sharing Group functionalities):

**New Sharing Group**

<table>
<thead>
<tr>
<th>General</th>
<th>Organisations</th>
<th>MISP Instances</th>
<th>Summary and Save</th>
</tr>
</thead>
</table>

**Name**

Financial Sector

**Releasable to**

Financial Sector organisations

**Description**

A general sharing group for the financial sector including financial sector actors like banks, insurance companies or payment processing companies.

- [x] Make the sharing group selectable (active)
- Next page
- **General**: Metadata describing the intent of the sharing group
  - **Name**: The unique name of the sharing group.
  - **Releasable to**: A human-readable description of who data marked with the sharing group is shareable with. This field is NOT used by MISP for anything besides for being an informational field aimed at extender organisations of the sharing group.
  - **Description**: A natural-text representation of the intent of the sharing group.
  - **Make the sharing group selectable (active)**: A sharing group can be made passive by unchecking this setting. All events and attributes will continue to adhere to a passive sharing group's distribution setting, however, the sharing group will not be offered as a selectable option when setting the distribution of events/attributes. The idea behind this is that ad-hoc sharing groups that have outlived their purpose can be retired in order to reduce the clutter in the UI.

### New Sharing Group

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>UUID</th>
<th>Extend</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>local</td>
<td>Org221</td>
<td></td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>remote</td>
<td>CIRCL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>remote</td>
<td>CthulhuSFRL.be</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Previous page] [Next page]
- **Organisations:** The second page of the tool contains the distribution list containing all organisations directly named as a member of the sharing group.

- **Add Local/remote organisations:** The organisations are split into two lists (shown as two tabs in the tool) for local and known remote/external organisations. Local organisations are expected to have at least one local user on the instance whilst remote organisations do not. Synchronising with remote instances will create remote organisations whenever a new event is received of a yet unknown organisation. Remote organisations can always be converted to local organisations - this becomes interesting if a user of an external organisation requests access to your MISP instance.

- **Extend checkmark:** Checking the extend checkmark makes the selected organisation an extender of the sharing group, meaning they can edit the sharing group. It is expected of these trusted partners that they adhere to the "releasable to" tag set on the general page. The organisation creating the sharing group is always included as an extender.

### New Sharing Group

<table>
<thead>
<tr>
<th>General</th>
<th>Organisations</th>
<th><strong>MISP Instances</strong></th>
<th>Summary and Save</th>
</tr>
</thead>
</table>

- **Enable roaming mode** for this sharing group (pass the event to any connected instance where the sync connection is tied to an organisation contained in the SG organisation list).

<table>
<thead>
<tr>
<th>Name</th>
<th>URL</th>
<th>All orgs</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local instance</td>
<td><a href="http://192.168.56.101">http://192.168.56.101</a></td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>other</td>
<td><a href="http://192.168.56.105">http://192.168.56.105</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Previous page] [Next page]
- **Servers:** The third page of the tool describes the MISP instances the data marked with the given sharing group are allowed to be synchronised with. Keep in mind that any user that can view an event on a given instance will have the right to pull the event to their home instance, as they are part of the sharing group, however the organisation distribution list will still apply.
- **Enable roaming mode:** This setting will disable the server list and rely purely on the organisation list to distribute the data. If a sync connection's host organisation is in the organisation distribution list the instance becomes eligible for synchronising the data marked with the sharing group. Generally this carries a slightly higher risk as it relies on administrators correctly setting up the host organisation settings, but it removes the need to know the specific instance urls where the event/attribute should flow.
- **Add instance:** Add an instance to the distribution list from the sync instances set up under sync actions -> servers
- **All orgs:** Checking this checkmark will automatically include all organisations on the given instance in the sharing group. This means that in order to exchange with all users of a linked community, one does not need to know every organisation residing on the instance. This also means that the distribution list will not include the organisation names, which can be interesting for certain privacy sensitive communities.

### New Sharing Group

<table>
<thead>
<tr>
<th>General</th>
<th>Organisations</th>
<th>MISP Instances</th>
<th>Summary and Save</th>
</tr>
</thead>
</table>

**General:** You are about to create the **Financial Sector** sharing group, which is intended to be releasable to **Financial Sector organisations**.

**Local organisations:** It will be visible to **all organisations on this instance**, from which **Org221** can extend the sharing group.

**External organisations:** It will also be visible to **CIRCL, CthulhuSPRL.be**, out of which **nobody** can extend the sharing group.

**Synchronisation:** Furthermore, events are automatically pushed to: **other**

You can edit this information by going back to one of the previous pages, or if you agree with the above mentioned information, click **Submit** to create the Sharing group.
Summary: Once everything is set up, MISP will summarise the sharing group in a highlighted text page, which is highly advised to be reviewed before submitting the new sharing group/editing the sharing group. Mistakes in the sharing group settings can lead to organisations that should not be involved in the sharing group getting access or organisations receiving unwanted editing rights to the sharing group. Keep in mind that even if you have submitted a sharing group, it is not propagated until an event/attribute receives the sharing group as the selected distribution.

Populate from Template

Templates allow users to rapidly populate events of a specific type by filling out a series of pre-defined fields. Users with template creation privileges can create new templates for their organisations or for all organisations on their instance. If you are interested in template creation, please refer to the templating section. For users trying to populate an event, after clicking on the populate from template button, you’ll be presented with a list of all currently accessible templates. Pick the one that best describes the event that you are creating.

| Choose element type | Phishing E-mail | Phishing E-mail with malicious attachment | Malware Report |
Once you have chosen a template, you'll be presented with the actual form contained within. Make sure you fill out as many fields as possible with the mandatory fields - marked by a star in a bracket such as this: (*) - are filled out. Templates are divided into sections, with each section having a title and a description in addition to a series of fields. Each field can be an attribute or a file attachment field. An attribute field has the following components:

<table>
<thead>
<tr>
<th>Field:</th>
<th>Artifacts Dropped (File) (*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>Insert any data you have on dropped files here.</td>
</tr>
<tr>
<td>Types:</td>
<td>filename</td>
</tr>
</tbody>
</table>

Describe the Artifacts Dropped (File) using one or several (separated by a line-break) of the following types: filename.
- **Field**: The name of the field along with an indication if the field is mandatory.
- **Description**: A short description of the field.
- **Types**: The value(s) that are valid for the field. In the case of several types being shown here, you can enter value(s) matching any one of the types, or in the case of a batch import field, any mixture of the given types.
- **Text field**: This field can either be a single line textfield or a multi-line text area. For the former, enter a single value of the above indicated type, whilst for the latter you can paste a list of values separated by line-breaks.

**Freetext Import Tool**

![Freetext Import Tool](image)
If you have a list of indicators that you would like to quickly generate attributes out of then the Free-text import tool is just what you need. Simply paste a list of indicators (separated by line-breaks into this tool).

**Freetext Import Results**

Below you can see the attributes that are to be created based on the results of the free-text import. Make sure that the categories and the types are correct, otherwise several options will be offered based on an automatic cluster resolution.

<table>
<thead>
<tr>
<th>Value</th>
<th>Category</th>
<th>Type</th>
<th>IDS</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.0.1</td>
<td>Network activity</td>
<td>p-cidt</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>domain.com</td>
<td>Network activity</td>
<td>hostname</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>chrome.exe</td>
<td>Payload delivery</td>
<td>f-ename</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>
Since there are several category / type combinations that can be valid for a lot of values, MISP will suggest the most common settings. You can alter the category / type / IDS fields manually if you disagree with the results. The options will be restricted to valid category/type combinations for the value that you have entered.

If any correlation is already found, these correlations will be displayed in the result page.

**Attribute Replace Tool**

If you would like to create and maintain an event with a set of indicators that receives removals and additions over time, then the attribute replace tool might make this task easier for you.
Simply select the desired category / type combination, choose whether the attributes should be marked for IDS exports and paste the new list of indicators into the textarea. Attributes of the same category/type that are present in the event but not the new list will be removed, values in the pasted list that do not yet exist as attributes will be created as attributes and values that already have matching attributes will be left untouched.

**Add attachments to the event**

You can also upload attachments, such as the malware itself, report files from external analysis or simply artifacts dropped by the malware. Clicking on the add attachment button brings up a form that allows you to quickly attach a file to the event. The following fields need to be filled out:

### Add Attachment

**Category**
- Antivirus detection

**Distribution**
- All communities

**Contextual Comment**

[Choose File] No file chosen

[ ] Malware

[Upload]
● **Category:** The category is the same as with the attributes, it answers the question of what the uploaded file is meant to describe.

● **Distribution:** This drop-down list allows you to control who will be able to see this attachment. The distribution is inherited by attributes: the most restrictive setting wins. For more info, refer to the distribution information in the event section.

● **Upload field:** By hitting browse, you can browse your file system and point the uploader to the file that you want to attach to the attribute. This will then be uploaded when the upload button is pushed.

● **Malware:** This check-box marks the file as malware and as such it will be zipped and passworded, to protect the users of the system from accidentally downloading and executing the file. Make sure to tick this if you suspect that the file is infected, before uploading it.

● **Contextual Comment:** You can add some comments to the attribute that will not be used for correlation but instead serves as purely an informational field.

**Propose a change to an event that belongs to another organisation**

If you would like to propose a modification to an attribute, or to propose some additional attributes to the creating organisation, you can do this with the buttons that replace the add attribute field on the left and the edit icon on the right end of each listed attribute in the event view. The creating organisation of the event will be able to see any proposals and discard or accept the changes.
If the organisation that has created the event is on another connected server, they will be able to accept the proposal once they initiate a pull and receive your proposal. After this they can republish the event, sending the altered attribute back to your instance.

**Populate from OpenIOC**

It is also possible to attempt to import the data contained in a .ioc file. The import tool will attempt to gather as many IndicatorItems within nested logical operators as possible without breaking their validity. After the procedure is done, you'll be presented with a list of successfully created attributes and a list of failed IndicatorItems as well as a graph of the .ioc file.

13 attributes created successfully, 6 indicators could not be mapped and saved.

### Successfully added attributes:

<table>
<thead>
<tr>
<th>Uuid</th>
<th>Category</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>b9ef1559-c059-46e3-81d9-52800545e16e</td>
<td>Other</td>
<td>other</td>
<td>FileItem:REInfo:Sections:Section/Name:stub</td>
</tr>
<tr>
<td>15604b6a-8281-4738-bfe8-bcc8117e8a38</td>
<td>Payload installation</td>
<td>filename</td>
<td>mdmcoo3.PNF</td>
</tr>
<tr>
<td>e07ad9d5b-9e6e-41ec-87ce-ee67338ed2e10</td>
<td>Payload installation</td>
<td>filename</td>
<td>mdmcoo3.PNF</td>
</tr>
<tr>
<td>e07ad9d5b-9e6e-41ec-87ce-ee67338ed2e10</td>
<td>Payload installation</td>
<td>filename</td>
<td>xen50.PNF</td>
</tr>
</tbody>
</table>
**Populate from ThreatConnect**

You can also import the data from a ThreatConnect export csv file. The following columns are used by the import tool (and are thus mandatory fields to select during the export):

- Type
- Value
- Confidence
- Description
- Source

The result will be a list of attributes that get added to the currently selected event, each of which will be marked with a comment that indicates that its origin being from a ThreatConnect import.

**Adding IOCs from a PDF report**

You can use a generic script called IOC parser or use a script published by Palo Alto to convert IOC parser output to a MISP event: [report_to_misp](https://github.com/PaloAltoNetworks-BD/report_to_misp).

**Publish an event**

- Publish Event
- Publish (no email)
- Contact Recorder
- Download as XML
- Download as IOC
Once all the attributes and attachments that you want to include with the event are uploaded / set, it is time to finalise its creation by publishing the event (click on publish event in the event view). This will alert the eligible users of it (based on the private-controls of the event and its attributes/attachments and whether they have auto-alert turned on), push the event to instances that your instance connects to and propagate it further based on the distribution rules. It also reads the network related attributes for NIDS signature creation (through the NIDS signature export feature, for more information, go to the export section.). There is an alternate way of publishing an event without alerting any other users, by using the "publish (no email)" button. This should only be used for minor edits (such as correcting a typo).

If your instance has background jobs enabled then the event might not get published immediately.

**Browsing past events**

The MISP interface allows the user to have an overview over or to search for events and attributes of events that are already stored in the system in various ways.

**To list all events**

On the left menu bar, the option "List events" will generate a list of the last 60 events. While the attributes themselves aren't shown in this view, the following pieces of information can be seen:
- **Published:** Already published events are marked by a checkmark. Unpublished events are marked by a cross.
- **Org:** The organisation that created the event.
- **Owner Org:** The organisation that owns the event on this instance. This field is only visible to administrators.
- **ID:** The event's ID number, assigned by the system when the event was first entered (or in the case of an event that was synchronized, when it was first copied over - more on *synchronisation* in chapter xy)
- **Tags:** Tags that are assigned to this event.
- **#Attr.:** The number of attributes that the event has.
- **Email:** The e-mail address of the event's reporter. This is not visible to regular users. Organisation administrators can see the e-mail addresses of their own organisation's users.
- **Date:** The date of the attack.
- **Threat Level:** The risk level of the attack, the following levels are possible:
  - **Low:** General Malware
  - **Medium:** Advanced Persistent Threats (APTs)
  - **High:** Sophisticated APTs and 0day exploits
  - **Undefined:** This field can be left undefined and edited at a later date.
- **Analysis:** Indicates the current stage of the analysis for the event, with the following possible options:
  - **Initial:** The analysis is just beginning
  - **Ongoing:** The analysis is in progress
  - **Completed:** The analysis is complete
- **Info:** A short description of the event, starting with an internal reference number.
- **Distribution:** This field indicates what the sharing privileges of the event. For details, refer to the distribution information in the event section.
- **Actions:** The controls that the user has to view or modify the event. The possible actions that are available (depending on user privileges - [click here](#) to find out more about privileges):
  - **Publish:** Publishing an event will have several effects: The system will e-mail all eligible users that have auto-alert turned on (and having the needed privileges for the event, depending on its private classification) with a description of your newly published event, it will be flagged as published and it will be pushed to all eligible servers (to read more about *synchronisation* between servers, have a look at the section on connecting servers)
  - **Edit:** Clicking on the edit button will bring up the same same screen as the one used for creating new events, with the exception that all fields come filled out with the data of the event that is being edited. The distribution of an event can only be edited if you are a user of the creating organisation of the event. For more information on this view, refer to the section on creating an event.
  - **Delete:** The system will prompt you before erasing the unwanted event.
  - **View:** Will bring up the event view, which besides the basic information contained in the event list, will also include the following:

**Filters**

It is also possible to filter the events shown by clicking on the small magnifying glass icons next to the field names and entering a filter term.

**Event view**
## Test Event 3

**Event ID**: 3  
**Org**: [John Doe](mailto:john.doe@example.com)  
**Tags**: [test](#)  
**Date**: 2014-03-05  
**Threat Level**: High  
**Analysis**: [test](#)  
**Distribution**: [test](#)  
**Description**: Test Event 3  
**Published**: Yes

### Related Events

- 2014-03-05 (1)  
- 2014-03-05 (2)  
- 2014-03-27 (4)

### Event Details

<table>
<thead>
<tr>
<th>Date</th>
<th>Category</th>
<th>Type</th>
<th>Value</th>
<th>Comment</th>
<th>Related Events</th>
<th>IDS</th>
<th>Distribution</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014-03-05</td>
<td>Network activity</td>
<td>q-src</td>
<td>1.1.1.1</td>
<td>An IP address</td>
<td>2.1</td>
<td>Yes</td>
<td>All</td>
<td><img src="#" alt="Actions" /></td>
</tr>
<tr>
<td>2014-03-05</td>
<td>q-src</td>
<td>2.2.2.2</td>
<td>An IP address</td>
<td>Yes</td>
<td>All</td>
<td><img src="#" alt="Actions" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014-03-05</td>
<td>q-src</td>
<td>3.3.3.3</td>
<td>An IP address</td>
<td>4</td>
<td>Yes</td>
<td>All</td>
<td><img src="#" alt="Actions" /></td>
<td></td>
</tr>
</tbody>
</table>
General Event Information

- **ID**: The ID of the event.
- **Uuid**: In order to avoid collisions between events and attributes (during for example a sync) a Uuid is assigned that uniquely identifies each of them.
- **Org**: The organisation that has originally created the event. The logo (if it exists on the server, alternatively a string) representing the organisation is also shown int he right upper corner.
- **Contributors**: Shows a list of the organisations that have contributed to the event via proposals. If you click any of the logos listed here, you'll get redirected to a filtered event history view, including only the changes made by the organisation.
- **Tags**: A list of tags associated with the event. Clicking a tag will show a list of events with the same tag attached. The little cross next to each tag allows you to remove the tag from the event, whilst the ‘+’ button allows you to assign a tag. For the latter two options to be visible, you have to have tagging permission.
- **Date**: The date of detection, set by the user that creates the event, not to be confused with the creation date of the event.
- **Threat Level**: The assigned threat level of the event.
- **Analysis**: The status of the analysis.
- **Distribution**: This shows the distribution rules applied to this event, controlling whether only the creating organisation can see (Your organisation only) it or everyone on the instance (This community only). The two remaining settings allow the event to be propagated to organisations on remote connected instances.
- **Info**: A short description of the event itself. Make sure not to put information in here that could be used for correlation purposes and be better suited as an Attribute.
- **Published**: Whether the event has been published or not. Publishing allows the attributes of the event to be used for all eligible exports and it notifies users that have subscribed to the event alerts. Also, a publish initiates a push to all eligible instances.

**List of Related Events** The list of relations is shown on the right hand side of the general event information. Events can be related by having one or more attributes that are exact matches. For example, if two events both contain a source IP attribute of 11.11.11.11 then they are related. The list of events that are related the currently shown one, are listed under "Related Events", as links (titled the related event's date and ID number) to the events themselves.

**Data Element Toggles** You can control some of the data that is shown on this page using three toggles. The elements that can be disabled are the pivot threads, the attributes (and proposals) and the Discussions. You can collapse these elements and then expand them again using the same button.

**Pivot Threads** While moving from event to event through the relation links (a process that we refer to as pivoting), you create a path that shows which events you have traversed. This path is reset by leaving the event view and navigating elsewhere in the application or by deleting the root pivot element. Each event visited is represented by a bubble in the pivot thread graph, connected by lines that show how the user has arrived at the next connected event. It is possible to jump back to an earlier relation and pivot to another event through that, creating branches in the graph. The currently selected event is coloured blue in the graph. If you would like to delete an element from the graph (including all of elements that branch off of it) just click on the small x within a pivot bubble. For a deletion to be possible the following conditions have to be met:

- The pivot element to be deleted cannot be on the path that leads to the currently selected event
- The pivot element residing in the graph's root can always be deleted - this will simply reset the current pivot thread

**Attributes and Proposals** A list of all attributes and proposals attached to the event. The fields for each of them only differ in the available actions and the fact that for proposals to attributes all fields are blank that would stay unchanged if the proposal was accepted (for example, proposing a change to an attribute to turn the IDS flag on will have all fields apart from the IDS flag blank in the proposal. Here is a list of what each of the fields represents:

- **Date**: The date of the last modification to the attribute. Proposals don't have a date of last edit.
- **Category**: The category of the attribute or proposal. For a list of possible categories visit the section on categories and types.
- **Type**: The type of the attribute or proposal. For a list of possible categories visit the section on categories and types.
- **Value**: The value or value-pair of the attribute. This is the main payload of the attribute, which is described by the category and type columns. For certain types of attributes that are made up of value-pairs the two parts will be split by a pipe (|), such as for filename|md5. The value field(s) are used by the correlation engine to find relations between events. In value-pair attributes both values are correlated individually.

- **Comment**: Attributes can have a contextual comment to further describe the attribute. These comments are not used for correlation and are purely informative.

- **Related Events**: A list of the event IDs that also contain an attribute with the same value.

- **IDS**: Flags an attribute as an indicator of compromise, allowing it to be included in all of the eligible exports.

- **Distribution**: Defines the distribution of the attribute individually. An attribute can have a different distribution level than the event. In any case, the lowest distribution level of the two is used.

- **Actions**: The user can interact with the events through these buttons, which will be further described in the next portion of the guide as they differ for attributes and proposals.

Depending on the colour coding of the row, you can have an attribute, a proposal to the event or a proposal to an attribute:

- **Attributes**: Each uncoloured line represents an Attribute.

- **Proposals to an Event**: Each gray line at the end of the list represents a Proposal to an event. These are proposals for a new attribute, mostly unrelated to any of the currently existing attributes. If the creator of the event accepts one of these a new attribute will be created.

- **Proposals to an Attribute**: Each attribute can have several edit proposals. These will be placed right below the attribute that the proposal affects and - as with the event proposals - is coloured grey. The original attribute's row is coloured blue if a proposal exists for it.

Using the modify button will bring up the attribute creation view, with all data filled out with the attribute's currently stored data.

**Event Discussion Thread**

Each event has its own assigned discussion where users (that are eligible to see the event) can participate in an open discussion. The users are anonymised in the messages, all that other users will see is their user ID number and their organisation. To post a message on the Event Discussion, either use the reply button on a previous post or use the quickresponse field at the bottom of the page. Each post is made up of the following:

- **Date**: The date when the post was created.

- **Post navigation**: This should the post's ID as well as a link to jump to the top of the discussion thread on the page itself.

- **Organisation logo**: If such an image exists for the organisation that has posted the message, then the logo is shown.

- **Message**: The body of the post itself. This can also include automatically generated links to other events and threads as well as show quoted text in embedded bubbles. Editing an event will also append a post with a message indicating that it was edited together with the timestamp of the edit.

- **User**: The e-mail address of the poster if he/she is from the organisation as the current user. Alternatively a generated sting is shown that includes the user ID of the user, so that his/her e-mail address could remain hidden whilst still being identifiable.

- **Action buttons**: Edit, Delete and Reply. The first two of the three options are only available to the poster of the message or a site admin. Quoting a post will automatically include the original message in [quote] tags.

Here is a list of the various tools you can use while using this feature:

- **Pagination**: There are 5 posts visible on each event page, if there have been more messages posted, use the previous and next button to navigate through the thread. This will not reload the rest of the page.

- **Discussion Tags**: Users can quote something by encapsulating it in [quote][/quote] tags, they can create a link to another event with the [event][/event] tags or to another discussion thread with [thread][/thread].

- **Quick Post**: Adding a post will take the user to a separate add Post page, something that can be a bit of an inconvenience. To avoid this, there is a quick post button, where users can add messages on the fly without having to reload the page. On top of the quick post field, 3 buttons allow users to generate quote, event and thread tags quickly.

**Event History**
View the logs of the event that show how the event has changed over time, including the contribution from other organisations in the form of proposals. There are two ways to get to this view, either by clicking on View Event History on the side menu of an event view, or by clicking on a contributing organisation's logo on the event view. The latter will show a restricted form of the logs, showing only Proposals created by the selected organisation. The fields shown in this view are as described as follows:

- **Org**: The logo (or in the lack thereof a string representation) of the organisation.
- **Action**: Each entry in the log happens during an action, such as the creation, modification or deletion of data and some special actions (such as accepting a proposal). This field shows which action caused the entry to be created.
- **Model**: As described above, a log entry is generated on certain actions. This field shows which type of data was affected that caused the log entry to be created (such as a change to the event, the creation of an attribute, the discarding of a proposal, etc).
- **Title**: This is a short description of the change itself and it is not nearly as detailed as the information administrators get in the audit logs. However, for attributes and proposals the category / type and value of the created or edited attribute is shown.
- **Created**: The date and time of the log entry's creation.

**Listing all attributes**

Apart from having a list of all the events, it is also possible to get a list of all the stored attributes in the system by clicking on the list attributes button. The produced list of attributes will include the following fields:

<table>
<thead>
<tr>
<th>Event</th>
<th>Org</th>
<th>Category</th>
<th>Type</th>
<th>Value</th>
<th>Comment</th>
<th>IDS</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Other</td>
<td>comment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Network activity</td>
<td>ip-src</td>
<td>2.2.2.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Network activity</td>
<td>ip-src</td>
<td>3.0.0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
● **Event**: This is the ID number of the event that the attribute is tied to. If an event belongs to your organisation, then this field will be coloured red.

● **Org**: The organisation that has created the event.

● **Category**: The category of the attribute, showing what the attribute describes (for example the malware's payload). For more information on categories, go to section xy

● **Type**: The type of the value contained in the attribute (for example a source IP address). For more information on types, go to section xy

● **Value**: The actual value of the attribute, describing an aspect, defined by the category and type fields of the malware (for example 11.11.11.11).

● **Comment**: An optional contextual comment attached to the attribute.

● **IDS**: Shows whether the attribute has been flagged for NIDS signature generation or not.

● **Actions**: A set of buttons that allow you to view the event that the attribute is tied to, to edit the attribute (using the same view as what is used to set up attributes, but filled out with the attribute's current data) and a delete button.

**Searching for attributes**

Apart from being able to list all events, it is also possible to search for data contained in the value field of an attribute, by clicking on the "Search Attributes" button.
This will bring up a form that lets you enter one or several search strings (separate search strings with line breaks) that will be compared to the values of all attributes, along with options to narrow down the search based on category and type. The entered search string has to be an exact match with (the sub-string of) a value. A second text field makes it possible to enter event IDs for events that should be excluded from the search (again, each line represents an event ID to be excluded). The third text field allows the user to restrict the results to attributes from certain organisations or to attributes not created by certain other organisations, using the above described syntax. The list generated by the search will look exactly the same as listing all attributes, except that only the attributes that matched the search criteria will be listed (to find out more about the list attributes view, click here). The search parameters will be shown above the produced list and the search terms will be highlighted. The last option is a checkbox that restricts all of the results to attributes that are marked as IDS signatures.

Attributes
Results for all attributes with the value containing "1.1.1".

<table>
<thead>
<tr>
<th>Event</th>
<th>Org</th>
<th>Category</th>
<th>Type</th>
<th>Value</th>
<th>Comment</th>
<th>IDS</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td>network</td>
<td>activity</td>
<td>1.1.1</td>
<td>1.1.1</td>
<td>IP address</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>network</td>
<td>activity</td>
<td>1.1.1</td>
<td>1.1.1</td>
<td>The same IP address</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Updating and modifying events and attributes

Every event and attribute can easily be edited. First of all it is important to find the event or attribute that is to be edited, using any of the methods mentioned in the section on browsing past events. Once it is found, the edit button (whether it be under actions when events/attributes get listed or simply on the event view) will bring up the same screen as what is used to create the entry of the same type (for an event it would be the event screen as seen here, for an attribute the attribute screen as described here). You can also simply double-click on the event you wish to edit and enter the edit mode. Keep in mind that editing any event (either directly or indirectly through an attribute) will unpublish it, meaning that you'll have to publish it (through the event view) again once you are done.

Tagging

As described earlier, users with tagging rights can arbitrarily tag events using tags chosen from a pool of available options. If you have tagging privileges and would like to create a new tag, navigate to Event Actions - Add Tag. You'll be presented with the following form:

**Add Tag**

<table>
<thead>
<tr>
<th>Name</th>
<th>Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSINT</td>
<td>#000000</td>
</tr>
</tbody>
</table>

Add
Fill out the following fields:

- **Name**: Pick a name for the tag. Try to use consistent naming conventions across your instance, to avoid confusion.
- **Colour**: You can choose a colour for the tag by clicking on the colour field and using the colour picker tool. Try to avoid having duplicate or similar looking colours to help avoid confusion.

### Templating

Newer users can easily be overwhelmed by having to manually populate events with attributes without any guidance. What sort of information should go into the event? What should be the category and type of a C2 IP? Templates allow users to use simple forms to populate events. Even though MISP ships with a few default templates, it is possible for users (with the appropriate templating privilege) to create new templates for their users or for all users of the instance. Let's look at how you can create a template. First go to Event Actions - Add Template to go to the event creation view.

#### Create Template

**Name**

OSINT Report

**Tags**

- [x] OSINT

**Event Description**

Use this template to create OSINT events.

- Share this template with others

[Create]
The following fields have to be filled out:

- **Name**: The name of the template should describe what type of an event it should be used to generate attributes.
- **Tags**: You can attach tags to the template - an event populated using the template would automatically receive the tag(s).
  Add new tags using the + button. If you change your mind about a tag you can remove it with the cross next to the tag name.
- **Event Description**: A short description about the events that this template should be used for.
- **Share this template with others**: The template can be set to be usable by any organisation on the instance or only by the one that has created it.

Once the skeleton template is created, you can start populating the template with data. There are 3 types of elements that can be used during the creation of a template: attribute, file and text elements. Text elements divide the template into sections with an information field, followed by all of the attribute/file fields until a new text field is read. Don't worry about the order of the elements during creation, they can be re-arranged using drag & drop. Let's look at the 3 element types:

**Attribute Element**

![Add Attribute Element To Template](image-url)
The following fields have to be filled out:

- **Name**: The field name that will be presented to the user.
- **Description**: A brief description of the element. Make sure that you provide sufficient information to the user to make it obvious what is expected.
- **Category**: The category used for any attributes created using this template element.
- **Type**: The type or complex type used for any attributes created using this template element. Complex types allow for several related types to be used on data entry. For example, a “file” complex type element allows for filenames and hashes.
- **Use Complex types**: If the category permits it, switch to a complex type using this checkbox.
- **Automatically mark for IDS**: If checked, any attributes generated using this element will be marked for IDS exporting.
- **Mandatory element**: If the element is marked as mandatory, then the template form can only be submitted by users if this field is filled out.
- **Batch import element**: Allow for multiple values to be entered (separated by line breaks).

**File Element**

![Add File Element To Template](image-url)

- **Name**: 
- **Description**: 
- **Category**
  - **Select Category**
  - Malware
  - Mandatory element
  - Batch import element
- **Submit**
The following fields have to be filled out:

- **Name**: The field name that will be presented to the user.
- **Description**: A brief description of the element. Make sure that you provide sufficient information to the user to make it obvious what is expected.
- **Category**: The category to be used by all attachments uploaded through this element.
- **Malware**: If the uploaded files are malicious and should be encrypted and password protected, mark this checkbox.
- **Mandatory element**: If it should be required to upload an attachment, check this checkbox.
- **Batch import element**: Ticking this checkbox allows users to upload several files using this element.

**Text Element**
The following fields have to be filled out:

- **Name**: The name of the section that will be presented to the user.
- **Text**: The description of the section. Explain briefly to the user what the following attribute/file elements will be dealing with. There are several ways to split a template into sections, try to have ease of use in mind while creating it.

## Contacting the reporter

To get in touch with the reporter of a previously registered event, just find the event for which you would like to contact the reporter by either finding it on the list of events, by finding it through one of its attributes or by finding it through a related event. Once the event is found and the event view opened, click the button titled "Contact Reporter". This will bring up a view where you can enter your message that is to be e-mailed to all members of the reporting organisation that subscribe to receiving such reports or the reporting user himself. Along with your message, the detailed information about the event in question will be included in the e-mail.
By default, the message will be sent to every member of the organisation that posted the event in the first place, but if you tick the check-box below the message field before sending the mail, only the person that reported the event will get e-mailed.

**Automation**

It is possible to quickly and conveniently export the data contained within the system using the automation features located in the main menu on the left (available to users with authentication key access only). There are various sets of data that can be exported, by using the authentication key provided by the system (also shown on the export page). If for whatever reason you would need to invalidate your current key and get a new one instead (for example due to the old one becoming compromised) just hit the reset link next to the authentication key in the export view or in your "my profile" view. To find out about the various export formats and the usage within the automation functions, please read the page on the API's usage.

**Exporting data**

For users that do not have authentication key access, an alternate export feature is available that relies on your interactive login to the site. To access these, just use the export menu button to the left and you’ll be presented with a list of export options.

Depending on your server’s configuration, you will be presented with one of two possible pages, depending on whether you have background processing enabled or not.

**Export page with background jobs disabled**

The page will list a set of export formats that you can immediately download as a file. Just click on the desired export format and MISP will start collecting all the data that you will receive in a file. Keep in mind that this can be a lengthy process. To avoid having to wait, consult with your instance’s site administrator about enabling the background processing.
Export page with background jobs enabled

If the background jobs are enabled, you'll be redirected to a different version of the export page. Here you will see a table with all of the major export formats and the current status of the cached export files. Keep in mind that these are generated on an organisation by organisation basis, so even though others have generated newer export caches your organisation may have an outdated cache. You can simply issue a generate command (by clicking the "Generate" button) on the desired export type and the background workers will start fetching and assembling your cache. A progress bar will show the progress of the export process.

Once done, you can click "Download" to download the freshly generated cache file. If the cache is already up to date from before, then you don't have to regenerate the cache, just click on the "download" button. You may have noticed that the TEXT export only has a generate button - this is because TEXT exports are made up of a lot of types of exports, all of which get generated together. To download any of these files, just click on any of the attribute types at the bottom of the table. A quick description of each of the fields in the table:

- **Type**: The type of the export (such as XML, Suricata, MD5, etc.).
- **Last Update**: The generation date of the current cache for the given export type.
- **Description**: A description of the export format.
- **Outdated**: This compares the cache generation date to the last timestamp when an event was updated and lets you know whether the cache is outdated or not.
- **Progress**: Shows the progress of the last initiated generation process.
- **Actions**: Download or Generate the given cache with these buttons.

Export

Export functionality is designed to automatically generate signatures for intrusion detection systems. To ensure signature generation for a given attribute, a signature file for this attribute must be set up. Note that not all attribute types are associated with signature generation currently. We only support HIDS signature generation for IP, domain names, user agents, etc., and that first 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:

<table>
<thead>
<tr>
<th>Type</th>
<th>Last Update</th>
<th>Description</th>
<th>Outdated</th>
<th>Progress</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML</td>
<td>N/A</td>
<td>Click this to download all events and attributes that you have access to the file containing these events in a custom XML format.</td>
<td>Yes</td>
<td>Completed</td>
<td>Download</td>
</tr>
<tr>
<td>CSV.5</td>
<td>N/A</td>
<td>Click this to download all attributes that are notifications and that you have access to events, where events contain numeric values in CSV format.</td>
<td>Yes</td>
<td>Completed</td>
<td>Download</td>
</tr>
<tr>
<td>CSV.5</td>
<td>N/A</td>
<td>Click this to download all attributes that you have access to the file containing these events in CSV format.</td>
<td>Yes</td>
<td>Completed</td>
<td>Download</td>
</tr>
<tr>
<td>Suricata</td>
<td>N/A</td>
<td>Click this to download all network-related attributes that you have access to the file containing these events. Only published events and attributes marked as Signature are exported.</td>
<td>Yes</td>
<td>Completed</td>
<td>Download</td>
</tr>
<tr>
<td>Snort</td>
<td>N/A</td>
<td>Click this to download all network-related attributes that you have access to the file containing these events. Only published events and attributes marked as Signature are exported.</td>
<td>Yes</td>
<td>Completed</td>
<td>Download</td>
</tr>
<tr>
<td>NIDS 2 weeks ago</td>
<td>N/A</td>
<td>Click one of these two buttons to download all NIDS checksums must be contained in the file containing these events. This list can be used to find forensic software when searching for suspicious files. Only published events and attributes marked as Signature are exported.</td>
<td>Yes</td>
<td>Completed</td>
<td>Download</td>
</tr>
<tr>
<td>SHA1</td>
<td>N/A</td>
<td>Click one of these two buttons to download all SHA1 checksums contained in the file containing these events. This list can be used to find forensic software when searching for suspicious files. Only published events and attributes marked as Signature are exported.</td>
<td>Yes</td>
<td>Completed</td>
<td>Download</td>
</tr>
<tr>
<td>TEXT</td>
<td>N/A</td>
<td>Click one of these two buttons to download all the attributes. This list can be used to find forensic software when searching for suspicious files. Only published events and attributes marked as Signature are exported.</td>
<td>Yes</td>
<td>Completed</td>
<td>Download</td>
</tr>
</tbody>
</table>
Exporting search results and individual events

Apart from the options offered by the export pages, it's also possible to export all events involved in a search attribute result table, by using the “Download results as XML” button on the left menu bar.

<table>
<thead>
<tr>
<th>Event</th>
<th>Category</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Network activity</td>
<td>ip-src</td>
<td>1.1.1.34</td>
</tr>
</tbody>
</table>

Page 1 of 1, showing 1 records out of 1 total, starting on record 1, ending on 1.
Each event’s view has its own export feature, both as an XML export and as a .ioc file. To reach these features, just navigate to an event and use the appropriate buttons on the right side.

<table>
<thead>
<tr>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results for all attributes with the value containing “1.1.1”:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event</th>
<th>Category</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Network activity</td>
<td>ip-src</td>
<td>1.1.1.34</td>
</tr>
</tbody>
</table>

Page 1 of 1, showing 1 records out of 1 total, starting on record 1, ending on 1
Connecting to other instances

Apart from being a self contained repository of attacks/malware, one of the main features of MISP is its ability to connect to other instances and share (parts of) its information. The following options allow you to set up and maintain such connections.

Setting up a connection to another server

In order to share data with a remote server via pushes and pulls, you need to request a valid authentication key from the hosting organisation of the remote instance. When clicking on List Servers and then on New Server, a form comes up that needs to be filled out in order for your instance to connect to it. The following fields need to be filled out:

<table>
<thead>
<tr>
<th>Base URL</th>
<th>Organization</th>
<th>Authkey</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://www.friendlymsp.com">https://www.friendlymsp.com</a></td>
<td>Org_name</td>
<td></td>
</tr>
</tbody>
</table>

- **PUSH**: ✔️
- **Pull**: ✔️

**Certificate file**: Choose File  
No file chosen

Add
- **Base URL**: The URL of the remote server.
- **Organization**: The organisation that runs the remote server. It is very important that this setting is filled out exactly as the organisation name set up in the bootstrap file of the remote instance.
- **Authkey**: The authentication key that you have received from the hosting organisation of the remote instance.
- **Push**: This check-box controls whether your server is allowed to push to the remote instance.
- **Pull**: This check-box controls whether your server can request to pull all data from the remote instance.
- **Self Signed**: Ticking this checkbox will allow syncing with instances using self-signed certificates.
- **Certificate File**: If the instance that you want to connect to has their entire own certificate chain, you can use this to import a .pem file with it and override CakePHP's standard root CA file.

If you are an administrator, trying to allow another instance to connect to your own, it is vital that two rules are followed when setting up a synchronisation account:

- The synchronisation user has to have the sync permission and full read/write/publish privileges turned on
- Both the sync user and the organisation setting in your instance’s Config/bootstrap.php file have to match the organisation identifier of the hosting organisation.

**Browsing the currently set up server connections and interacting with them**

If you ever need to change the data about the linked servers or remove any connections, you have the following options to view and manipulate the server connections, when clicking on List Servers: (you will be able to see a list of all servers that your server connects to, including the base address, the organisation running the server the last pushed and pulled event IDs and the control buttons).

![Servers](image-url)
• **Editing the connection to the**: By clicking edit a view, that is identical to the new instance view, is loaded, with all the current information of the instance pre-entered.

• **Deleting the connection to the instance**: Clicking the delete button will delete the link to the instance.

• **Push all**: By clicking this button, all events that are eligible to be pushed on the instance you are on will start to be pushed to the remote instance. Events and attributes that exist on the far end will be updated.

• **Pull all**: By clicking this button, all events that are set to be pull-able or full access on the remote server will be copied to this instance. Existing events will not be updated.

**Rest API**

The platform is also RESTful, so this means that you can use structured format (XML or JSON) to access Events data.

**Requests**

Use any HTTP compliant library to perform requests. You can choose which format you would like to use as input/output for the REST calls by specifying the Accept and Content-Type headers.

The following headers are required if you wish to receive / push XML data: **Authorization: your authorisation key** **Accept:** application/xml **Content-Type: application/xml**

The following headers are required if you wish to receive / push JSON data: **Authorization: your authorisation key** **Accept:** application/json **Content-Type: application/json** The following table shows the relation of the request type and the resulting action:

<table>
<thead>
<tr>
<th>HTTP format</th>
<th>URL</th>
<th>Controller action invoked</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/events</td>
<td>EventsController::index()</td>
</tr>
<tr>
<td>GET</td>
<td>/events/123</td>
<td>EventsController::view(123)</td>
</tr>
<tr>
<td>POST</td>
<td>/events</td>
<td>EventsController::add()</td>
</tr>
<tr>
<td>PUT</td>
<td>/events/123</td>
<td>EventsController::edit(123)</td>
</tr>
<tr>
<td>DELETE</td>
<td>/events/123</td>
<td>EventsController::delete(123)</td>
</tr>
<tr>
<td>POST</td>
<td>/events/123</td>
<td>EventsController::edit(123)</td>
</tr>
</tbody>
</table>

*Attachments are included using base64 encoding below the data tag.

**Example - Get single Event**

In this example we fetch the details of a single Event (and thus also his Attributes). The request should be:

```plaintext
GET https://your_misp_url/events/123
```

And with the HTTP Headers: **Accept:** application/xml **Authorization:** your_api_key

The response you're going to get is the following data:

```xml
<response>
  <Event>
    <id>57</id>
    <org>NCIRC</org>
    <date>2014-03-04</date>
    <threat_level_id>1</threat_level_id>
    <info>Code monkey doing code monkey stuff</info>
    <published>1</published>
  </Event>
</response>
```

**Using the System**
Example - Add new Event

In this example we want to add a single Event. The request should be:

POST https://your_misp_url/events
Accept: application/xml
Authorization: your_api_key

And the request body:

<Event>
  <date>2014-03-04</date>
  <threat_level_id>1</threat_level_id>
  <info>Something concise</info>
  <published>1</published>
  <analysis>1</analysis>
  <distribution>1</distribution>
  <Attribute>
    <type>other</type>
    <category>Artifacts dropped</category>
    <to_ids>1</to_ids>
    <event_id>57</event_id>
    <timestamp>1393327600</timestamp>
    <comment>This is an Attribute</comment>
    <value>Some_attribute</value>
  </Attribute>
</Event>

The response you're going to get is the following data:

HTTP/1.1 100 Continue
HTTP/1.1 200 Continue
Date: Tue, 04-Mar-2014 15:00:00
Server: Apache/2.2.22 (Ubuntu) PHP/5.4.9-4ubuntu2.3
X-Powered-By: PHP/5.4.9-4ubuntu2.3
Set-Cookie: CAKEPHP=deleted; expires=Wed, 05-Mar-2014 15:00:00 GMT; path=/
Set-Cookie: CAKEPHP=a4ok3lr5p9n5drgqj27625i4le3; expires Tue, 04-Mar-2014 15:00:00 GMT; path=/; HttpOnly
Content-Length: 1 kB
Content-Type: application/xml

<?xml version = "1.0" encoding = "UTF-8"?>
<response>
  <Event>
    <id>76</id>
    <org>NCIRC</org>
    <date>2014-03-04</date>
    <threat_level_id>1</threat_level_id>
    <info>Something concise</info>
    <published>1</published>
    <uuid>50aa54aa-f7a0-4d74-928d-18f0ff32448e</uuid>
    <attribute_count>1</attribute_count>
    <timestamp>1393328991</timestamp>
    <distribution>1</distribution>
    <proposal_email_lock>0</proposal_email_lock>
    <org>iglocska</org>
    <locked>0</locked>
    <publish_timestamp>1393947960</publish_timestamp>
    <Attribute>
      <id>10462</id>
      <type>other</type>
      <category>Artifacts dropped</category>
      <to_id>3</to_id>
      <uuid>50aa54bb-adec-4544-b412-18f0ff32448e</uuid>
      <event_id>76</event_id>
      <distribution>1</distribution>
      <timestamp>1393328991</timestamp>
      <comment/>
      <value>Some_attribute</value>
    </Attribute>
    <ShadowAttribute/>
  </RelatedEvent>
  <RelatedEvent>
    <id>76</id>
    <org>NCIRC</org>
    <date>2012-11-19</date>
    <info>Code monkey doing code monkey stuff</info>
    <uuid>50aa54aa-f7a0-4d74-918d-18f0ff32448e</uuid>
    <published>1</published>
    <analysis>1</analysis>
    <attribute_count>1</attribute_count>
    <org>iglocska</org>
    <timestamp>1393327600</timestamp>
    <distribution>1</distribution>
    <proposal_email_lock>0</proposal_email_lock>
    <locked>0</locked>
    <threat_level_id>1</threat_level_id>
    <publish_timestamp>1393947655</publish_timestamp>
  </RelatedEvent>
</Event>
<xml_version>2.2.0</xml_version>
</response>

The response from requesting an invalid page

<?xml version = "1.0" encoding = "UTF-8"?>
<response>
  <name>Not Found</name>
  <url>The_meaning_of_life</url>
</response>
Delegation

In information sharing, privacy of the reporting organisation can be important in such case as:

- an incident doesn't want to be linked to a potential victim.
- to avoid the relation of an organisation with the information shared.

MISP has a functionality to delegate the publication and completely remove the binding between the information shared and its organisation. If you want to publish an event without you or your organisation being tied to it, you can delegate the publication to an other organisation. That also means they will take the ownership of the event.

[warning] You need to have a role with "Delegation access" to delegate an event.
[warning] Also activate MISP.delegation parameter in your instance.

Send a delegation request

To do so, you first need to put the distribution of the event as "your organisation only".

<table>
<thead>
<tr>
<th>Publish Event</th>
<th>Publish (no email)</th>
<th>Delegate Publishing</th>
<th>Contact Reporter</th>
<th>Download as...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution</td>
<td>Your organisation only</td>
<td>Delegate me</td>
<td>Published</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Otherwise the delegation option will not be available.

<table>
<thead>
<tr>
<th>Publish Event</th>
<th>Distribution</th>
<th>Published</th>
<th>Info</th>
<th>Activity</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publish (no email)</td>
<td>All communities</td>
<td>No</td>
<td>Delegate me</td>
<td>0 (0)</td>
<td>Enabled (disable)</td>
</tr>
</tbody>
</table>
When the "Delegate Publishing" option is clicked, a pop-up will show up:

**Delegate the publishing of the Event to another organisation**

*Warning: You are about to request another organisation to take ownership of this event.*

**Target Organisation**
- Select organisation

**Desired Distribution**
- Recipient decides

**Message to the recipient organisation**

[Yes] [No]
Here you can choose

- to which organisation you wish to delegate the event among all those registered on the server. For this example we are going to ask Setec Astronomy to publish the event for us.
- The distribution option you would like to put on the event. You can let the other organisation (called "recipient") choose if you don't mind it. For this example, we will request the recipient to share it to all communities, but it is only a suggestion, and the recipient will be able to modify the diffusion setting if wanted.
Finally you can leave a free message to the recipient organisation.
Once the request is sent, a message will appear on the event to remind you of your request.

<table>
<thead>
<tr>
<th>Published</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sightings</td>
<td>0 (0)</td>
</tr>
<tr>
<td>Activity</td>
<td></td>
</tr>
<tr>
<td>Delegation request</td>
<td>You have requested that Setec Astronomy take over this event (View request details)</td>
</tr>
<tr>
<td>Correlation</td>
<td>Enabled (disable)</td>
</tr>
</tbody>
</table>
You can also see more details by clicking on “View request details”

Event Delegation

Request details

Your organisation is requesting Setec Astronomy to take over this event.

Message from requester

Discard  Cancel
And you can also discard the request yourself, by using this pop-up or the link in the left menu.

**Answer a delegation request**

As the recipient organisation, you will then receive the request of delegation. You will be notified by a red circle around the envelope on the top right of the screen.
When you click it, you will be redirected as usual on the dashboard, where we can see one delegation request on the left frame.

**Dashboard**

**Notifications**
- Proposals: 0 (View)
- Events with proposals: 0 (View)
- Delegation requests: 1 (View)

**Changes since last visit**
- Events updated: 571 (View)
- Events published: 571 (View)

Reset
Clicking on the "view" link then redirect to an event list view showing all the events other organisations wish to delegate to your organisation. Here we only see one event, from Acme Factory.

**Events**

<table>
<thead>
<tr>
<th>Published</th>
<th>Org</th>
<th>Id</th>
<th>Clusters</th>
<th>Tags</th>
<th>#Attr.</th>
<th>#Sightings</th>
<th>#Posts</th>
<th>Date</th>
<th>Threat Level</th>
<th>Analysis</th>
<th>Info</th>
<th>Distribution</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>×</td>
<td>Acme Factory</td>
<td>1096</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>2017-03-23</td>
<td>High</td>
<td>Initial</td>
<td>Delegate</td>
<td>Organisation</td>
<td>Not published</td>
</tr>
</tbody>
</table>
And here are the metadata of the so called event.

<table>
<thead>
<tr>
<th>Delegate me</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event ID</td>
</tr>
<tr>
<td>Uuid</td>
</tr>
<tr>
<td>Org</td>
</tr>
<tr>
<td>Contributors</td>
</tr>
<tr>
<td>Tags</td>
</tr>
<tr>
<td>Date</td>
</tr>
<tr>
<td>Threat Level</td>
</tr>
<tr>
<td>Analysis</td>
</tr>
<tr>
<td>Distribution</td>
</tr>
<tr>
<td>Info</td>
</tr>
<tr>
<td>Published</td>
</tr>
<tr>
<td>Sightings</td>
</tr>
<tr>
<td>Activity</td>
</tr>
<tr>
<td>Delegation request</td>
</tr>
<tr>
<td>Correlation</td>
</tr>
</tbody>
</table>
You will be able to view the details by clicking the so-called link.

Event Delegation

Request details

*Acme Finance* is requesting your organisation to take over this event.
The desired distribution level is *All communities*

Message from requestor

This is a delegation test

[Accept]  [Discard]  [Cancel]
If your role have publishing rights, you will be able to manage the delegation request by using one of the two links in the left menu.

You can either discard it:

![Delete Delegation Request](image)

Are you sure you would like to discard the request by Acme Finance to take ownership of Event #1113?

Yes  No
Or accept the delegation:

**Accept Delegation Request**

Are you sure you would like to accept the request by Acme Finance to take ownership of Event #1113?

[Yes] [No]
Please notice that the distribution desired by the requester will not automatically be set on the event, which will stay as distributed to your own organisation only if the parameter is not modified.
MISP Extended Events

MISP can now extend an event (starting from version 2.4.90). This allows users to build full blown events that extend an existing event, giving way to a combined event view that includes a sum total of the event along with all extending events. More
Administration

Users
- Adding a new user
- Listing all users
- Contacting a user

Organisations
- Adding a new organisation
- Listing all organisations
- Merge organisations

Roles
- Adding a new role
- Listing roles

Tools

Server Settings
- Server settings and diagnostics
  - Worker types
  - Workers dead

Import Blacklist
- Adding and modifying entries

Import Regexp
- The purpose of Import Regexp entries
- Adding and modifying entries

Managing the Signature whitelist
- Whitelisting an address
- Managing the list

Using MISP logs
- Browsing logs
- Searching Logs

Background Processing
- Command Line Tools for the Background Workers
- Monitoring the Background Processes
- Scheduling Jobs and Recurring Jobs

Various administration tips & tricks
- Setting a Publish Alert Filter
- Default sharing level
- Adding organisation logos
- The _schdlr_ worker is not starting
- How to redirect HTTP to HTTPS
- Increase max size of Samples / other files
- Support & feature requests
- More information in the notification emails about new events
- Get top API users
- MISP Logs
- Logging of failed authentication attempts
- Clearing expired sessions
- Troubleshooting MISP not connecting to redis but redis-cli working
- Errors about fields or tables

Jobs ~ TODO: Explain differences Default, Email, Cache

Scheduled Tasks
Administration

- Users
- Organisations
- Roles
- Tools
- Server Settings
- Jobs
- Scheduled Tasks

[warning] This page is under modification for updating the content. Current status:

- [x] Users - Reviewed/Updated on: ?
- [x] Organisations - Reviewed/Updated on: ?
- [x] Roles - Reviewed/Updated on: ?
- [x] Tools - Reviewed/Updated on: ?
- [ ] Server Settings - Reviewed/Updated on: ?
- [ ] Jobs aka. Background processing - Reviewed/Updated on: ?
- [ ] Scheduled Tasks aka. Background processing - Reviewed/Updated on: ?

Users

As an admin (not to be confused with Org Admin), you can set up new accounts for users, edit user profiles, delete them, or just have a look at all the viewers’ profiles. Organisation admins (Org Admin) are restricted to executing these actions exclusively within their own organisation’s users only.

Adding a new user

To add a new user, click on the Add User button in the administration menu to the left and populate the fields available the loaded view:
Admin Add User

Email

Set password

Password

Confirm Password

Organisation

Role

Choose organisation

admin

Authkey

Nids Sid

Sync user for

Not bound to a server

GPG key

Fetch GPG key

Receive alerts when events are published

Receive alerts from "contact reporter" requests

Disable this user account

Submit
Email: The user’s e-mail address, this will be used as his/her login name and as an address to send all automated e-mails as well as e-mails sent by contacting the user as the reporter of an event.

Set password: Tick the box if you want to define a temporary user-password for the user. If you don't, you should use the action button 'reset password' in the 'List Users' view to generate one and send it by email to the user.

Password: This textbox is displayed only when 'Set password' is ticked. A Temporary password for the user that he/she should change after the first login. Ensure that password pass the MISP password policy.

Confirm Password: This textbox is displayed only when 'Set password' is ticked. This should be an exact copy of the Password field.

Organisation: A drop-down list enables you to choose an organisation for the user. To learn more about organisation, click here.

Roles: A drop-down list allows you to select a role-group that the user should belong to. Roles define user privileges attributed to the user. To learn more about roles, click here.

Authkey: This is assigned automatically and is the unique authentication key of said user (he/she will be able to reset this and receive a new key). It is used for exports and for connecting one server to another, but it requires the user to be assigned to a role that has auth permission enabled.

NIDS Sid: ID of network intrusion detection systems.

Sync user for: Use this option for granting the user the right to synchronize the event between MISP server. This option is available for admin, Org Admin and Sync user role.

Gpgkey: The key used to encrypt e-mails sent through the system.

Fetch GnuPG key: Fetch GnuPG public key.

Receive alerts when events are published: This option will subscribe the new user to automatically generated e-mails whenever an event is published.

Receive alerts from "contact reporter" requests: This option will subscribe the new user to e-mails that are generated when another user tries to get in touch with an event's reporting organisation that matches that of the new user.

Disable this user account: Tick it if you want to disable this user account. (preferred to removing an account)

Listing all users

To list all current users of the system, just click on List Users under the administration menu to the left. A view will load containing a list of all users and the following columns of information:

<table>
<thead>
<tr>
<th>ID</th>
<th>Org</th>
<th>Role</th>
<th>Email</th>
<th>Authkey</th>
<th>Authkey</th>
<th>Contactkey</th>
<th>Gpgkey</th>
<th>Misp Sid</th>
<th>Termaccepted</th>
<th>Last login</th>
<th>Disabled</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>CIRCL</td>
<td>Publisher</td>
<td><a href="mailto:opensirt@opsiru.eu">opensirt@opsiru.eu</a></td>
<td>196359ea26e253a3f53c7f3d2f35d19e90f5f4b9 7cbb1 6b2</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>123456</td>
<td>Yes</td>
<td>2016-08-04</td>
<td>No</td>
<td>![ ]</td>
</tr>
<tr>
<td>1</td>
<td>MISP</td>
<td>admin</td>
<td><a href="mailto:admin@opsiru.eu">admin@opsiru.eu</a></td>
<td>dce9a862f3c7f61f90f5f4b97cbb16b2</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>400000</td>
<td>Yes</td>
<td>2016-08-04</td>
<td>No</td>
<td>![ ]</td>
</tr>
<tr>
<td>2</td>
<td>MISP</td>
<td>Publisher</td>
<td><a href="mailto:opensirt@OPSIRU.ORG">opensirt@OPSIRU.ORG</a></td>
<td>6f7338f3c7f61f90f5f4b97cbb16b2</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>100000</td>
<td>Yes</td>
<td>2016-08-04</td>
<td>No</td>
<td>![ ]</td>
</tr>
</tbody>
</table>
- **Id**: The user's automatically assigned ID number.
- **Org**: The organisation that the user belongs to.
- **Email**: The e-mail address (and login name) of the user.
- **Authkey**: Unique authentication key of the user.
- **Autoalert**: Shows whether the user has subscribed to auto-alerts and is continuing to receive mass-emails regarding newly published events that he/she is eligible for.
- **Contactalert**: Shows whether the user has the subscription to contact reporter e-mails directed at his/her organisation is turned on or off.
- **Gpgkey**: Shows whether the user has entered a GnuPG key yet.
- **Nids Sid**: Displays the currently assigned NIDS ID.
- **Termsaccepted**: This flag indicates whether the user has accepted the terms of use or not.
- **Last login**: Date of last login.
- **Disabled**: Displays the user status. Enabled or disabled.
- **Action Buttons**: There are 4 options available: reset the password, edit the user, delete the user or display a user's information. These options are also available on the left menu.
  - **Reset Password**: Use this action to reset a password. If you've created a new user without a password, tick the 'First time registration' checkbox to send a welcome message. Otherwise a reset password message will be sent.
- **Edit the user**: Same options of create user's view. Only a few options are available here:
  - **Terms accepted**: Indicates whether the user has accepted the terms of use already or not.
  - **Change Password**: Setting this flag will require the user to change password after the next login.
  - **Reset Auth Key**: Use this link for generate a new AuthKey.
Delete the user: If you want to delete a user. (Note: disabling is the preferred method)

Are you sure you want to delete #57? It is highly recommended to never delete users but to disable them instead.
- **Display the user**: Display all user's information.

### User

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>2</td>
</tr>
<tr>
<td>Org</td>
<td>MSP</td>
</tr>
<tr>
<td>Role</td>
<td>Publisher</td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:user@misp.training">user@misp.training</a></td>
</tr>
<tr>
<td>Autoalert</td>
<td>No</td>
</tr>
<tr>
<td>ContactMethod</td>
<td>No</td>
</tr>
<tr>
<td>Authtoken</td>
<td>sCv1QrV7MqKtjV2jYlY9h8Kx5FwJpZM+y5Q6C</td>
</tr>
<tr>
<td>Invited By</td>
<td><a href="mailto:admin@misp.training">admin@misp.training</a></td>
</tr>
<tr>
<td>PGP Key</td>
<td>NA</td>
</tr>
<tr>
<td>Note Flag</td>
<td>100000</td>
</tr>
<tr>
<td>Terms accepted</td>
<td>Yes</td>
</tr>
<tr>
<td>Password change</td>
<td>No</td>
</tr>
<tr>
<td>Newman</td>
<td>2014/08/04 10:18:39</td>
</tr>
<tr>
<td>Disabled</td>
<td>No</td>
</tr>
</tbody>
</table>

### Related Events
Contacting a user

Site admins can use the "Contact users" feature to send all or individual user an e-mail. Users that have a GnuPG key set will receive their e-mails encrypted. When clicking this button on the left, you'll be presented with a form that allows you to specify the type of the e-mail, who it should reach and what the content is using the following options:

**Contact User(s)**

**Messaging - here's a quick guide on how this feature works**

You can use this view to send messages to your current or future users or send them a temporary password.

- When adding a new user to the system, or when you want to manually reset the password for a user, just use the "Send temporary password" setting.
- After selecting the action, choose who the target of the e-mails should be (all users, a single user or a user not yet in the system).
- You can then specify (if eligible) what the e-mail address of the target is (for existing users you can choose from a dropdown menu).
- In the case of a new user, you can specify the future user's gpg key, to send their new key in an encrypted e-mail.
- The system will automatically generate a message for you, but it is also possible to write a custom message if you tick the check box, but don't worry about assigning a temporary password manually, the system will do that for you, right after your custom message.

<table>
<thead>
<tr>
<th>Action</th>
<th>Subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom message</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recipient</th>
<th>Recipient Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>A single user</td>
<td><a href="mailto:admin@hisp.nl">admin@hisp.nl</a></td>
</tr>
</tbody>
</table>

- Enter a custom message

Message

Submit
- **Action:** This defines the e-mail type, which can be a custom message or a password reset. Password resets automatically include a new temporary password at the bottom of the message and will automatically change the user's password accordingly.
- **Subject:** In the case of a custom e-mail, you can enter a subject line here.
- **Recipient:** The recipient toggle lets you contact all your users, a single user (which creates a second drop-down list with all the e-mail addresses of the users) and potential future users (which opens up a text field for the e-mail address and a text area field for a GnuPG public key).
- **Custom message checkbox:** This is available for password resets and for welcome messages. You can either write your own message (which will be appended with a temporary key and the signature), or let the system generate one automatically.

Keep in mind that all e-mails sent through this system, in addition to your own message, will be signed in the name of the instance's host organisation's support team, the e-mail will also include the e-mail address of the instance's support (if the contact field is set in the bootstrap file), and will include the instance's GnuPG signature for users that have a GnuPG key set (and thus are eligible for an encrypted e-mail).
GnuPG instance key is the GnuPG key used by the MISP instance and which is only used to sign notification. The GnuPG key used in the MISP instance must not be used anywhere else and should not be valuable.

Organisations

Each user belongs to an organisation. As admin, you can manage these organisations.

Adding a new organisation

To add a new organisation, click on the "Add Organisation" button in the administration menu to the left and fill out the following fields in the view that is loaded:
- **Local organisation**: If the organisation should have access to this instance, tick the checkbox. If you would only like to add a known external organisation for inclusion in sharing groups, uncheck it.
- **Organisation Identifier**: Name your organisation. If you want to add a picture, you should add a file on the webserver using the 'Server Settings menu'. Picture should have the same name. To learn more about server settings menu, [click here](#).
- **Uuid**: Unique identifier. If you want to share organisation between MISP multi-instance, use the same Uuid.
- **A brief description of the organisation**: A word for describing the organisation.
- **Nationality**: A drop-down list for selecting the country of organisation.
- **Sector**: Define the sector of organisation (financial, transport, telecom...)
- **Type of organisation**: Define the type of the organisation.
- **Contacts**: You can add some contact details for the organisation.

**Listing all organisations**

To list all current organisations of the system, just click on List Organisations under the administration menu to the left. There are 3 tabs in this view to filter local organisations, remote organisations or both. The default view displays local organisations. For all views the following columns of information are available:
- **Id**: The organisation's automatically assigned ID number.
- **Logo**: Picture of the organisation.
- **Name**: Name of the organisation.
- **Uuid**: Unique identifier of organisation. Share this Uuid when using it between MISP's multi-instance.
- **Description**: Description of the organisation.
- **Nationality**: Country of the organisation.
- **Sector**: Sector defined for the organisation.
- **Type**: Type of organisation.
- **Contacts**: Contacts of organisation.
- **Added by**: Login of the user who added the organisation
- **Local**: Flag defined if the organisation is local or remote.
- **Users**: The amount of users on this instance belonging to the organisation.
- **Actions**: There are 3 options available: edit, delete or display an organisation's information. These options are also available on the left menu when you are on the display view.
  - **Edit Organisation**: Same options of create organisation's view.
- **Delete Organisation:** Use this option for deleting organisation.

  Are you sure you want to delete MISP?

  [OK]  [Cancel]
- **View Organisation**: Use this option to display information about the selected organisation. In this view, you can display the user belongs to this organisation and events published by organisation.
Merge organisations

Merge Organisation menu is available only in the organisation view, under the left menu. Merging one organisation into another will transfer all users and data from one organisation to a different one. The organisation of which the users and data will be transferred is displayed on the left, the target organisation is displayed on the right.
Roles

Privileges are assigned to users by assigning them to rule groups. Rule groups use one of four options determining what they can do with events as well as four additional privilege elevation settings. These are the four options to edit the full options available in the Roles section: Read Only, Manage My Own Events, Manage Organisation Events, Manage & Publish Organisation Events. A short description is provided below:

- **Read Only**: Allows a user to browse events that his organisation has access to, but doesn't allow any changes to be made to the database.
- **Manage My Own Events**: Allows users to create, modify or delete their own events, but they cannot publish them.
- **Manage Organization Events**: Allows users to create events or modify and delete events created by a member of their organisation.
- **Manage & Publish Organisation Events**: Gives users the right to do all of the above and to publish the events of their organisation.

The extra permissions are defined below:

- **Perm Admin**: Gives the user limited administrator privileges, this setting is used for an organisation's admins.
- **Perm Audit**: Grants access to the logs. With the exception of site admins, only logs generated by the user's own org are visible.
- **Perm Tagger**: Allows a user to assign tags to events.
- **Perm Sharing Group**: Grant access to edit or create sharing groups.
- **Perm Site Admin**: Gives the user full administrator privileges, this setting is used for site admins.
- **Perm Auth**: This setting enables the authentication key of the role's users to be used for rest requests.
- **Perm Tag Editor**: Grants access to edit or create tags.
- **Perm Delegate**: Grant access to delegate the publication of an event to a third-party organization.
- **Perm Sync**: This setting enables the users of the role to be used as a synchronisation user. The authentication key of this user can be handed out to the administrator of a remote MISP instance to allow the synchronisation features to work.
- **Perm Regexp Access**: Allows users who have this permission enabled to edit the regular expression table. Be careful when giving out this permission, incorrect regular expressions can be very harmful (infinite loops, loss of data, etc.).
- **Perm Template**: Grant access to create or modify templates.

Adding a new role

When creating a new role, you will have to enter a name for the role to be created and set up permissions (as described above) using the drop-down menu and related check-boxes.

Add Role

<table>
<thead>
<tr>
<th>Name</th>
<th>Permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perm Admin</td>
<td>Read Only</td>
</tr>
<tr>
<td>Perm Audit</td>
<td>Perm Site Admin</td>
</tr>
<tr>
<td>Perm Tagger</td>
<td>Perm Auth</td>
</tr>
<tr>
<td>Perm Sharing Group</td>
<td>Perm Tag Editor</td>
</tr>
<tr>
<td>Perm Delegate</td>
<td>Perm Sync</td>
</tr>
<tr>
<td>Perm Regexp Access</td>
<td>Perm Template</td>
</tr>
</tbody>
</table>
Listing roles

By clicking on the List Roles button, you can view a list of all currently registered roles and their enabled permissions. In addition, you can find buttons that allow you to edit and delete said roles. Keep in mind that you will need to first remove every member from a role before you can delete it.

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Permissions</th>
<th>Admin</th>
<th>Site Admin</th>
<th>Sync</th>
<th>Add Event</th>
<th>Edit Event</th>
<th>Add Tag</th>
<th>Tag Editor</th>
<th>Template Editor</th>
<th>Sharing Editor</th>
<th>Delegations Editor</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Admin</td>
<td>Manage &amp; Public Organization Events</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>2</td>
<td>User</td>
<td>Manage Organizations Events</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>3</td>
<td>Read Only</td>
<td>Read Only</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

Page 1 of 1, showing 7 records out of 7 total, starting on record 1, ending on 7
- **Id**: The role's automatically assigned ID number.
- **Name**: The name of role.
- **Permission**: One of the 4 permissions: Read Only, Manage My Own Events, Manage Organization Events, Manage & Publish Organization Events.
- **Extra Permissions flag**: Flag for each extra permissions: Admin, Site Admin, Sync Actions, Audit Actions, Auth key access, Regex Actions, Tagger, Tag Editor, Template Editor, Sharing Group Editor, Delegations Access.
- **Action Buttons**: There are 2 options available: Edit Role or Delete it.
  - **Edit Role**: Same options of create role's view.

<table>
<thead>
<tr>
<th>Name</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automation user</td>
<td>Manage &amp; Publish Organization</td>
</tr>
<tr>
<td>Perm Admin</td>
<td>Perm Site Admin</td>
</tr>
<tr>
<td>Perm Audit</td>
<td>Perm Auth</td>
</tr>
<tr>
<td>Perm Tagger</td>
<td>Perm Tag Editor</td>
</tr>
<tr>
<td>Perm Sharing Group</td>
<td>Perm Delegate</td>
</tr>
</tbody>
</table>

[Image of role UI]
• 
  • **Delete Role:** Use this option to delete a role.

Are you sure you want to delete Read Only?

[OK]  [Cancel]
Tools

MISP has a couple of administrative tools that help administrators keep their instance up to date and healthy. The list of these small tools can change rapidly with each new version, but they should be self-explanatory. Be sure to check this section after each upgrade to a new version, just in case there's a new upgrade script in there - though if this is the case it will be mentioned in the upgrade instructions.

Administrative actions

- Reattach destination asset events
- Reattach destination relationship attributes
- Reset the attribute counts (events need to have no validation issues)
- Reattach attributes
- Reattach proposals
- Verify DNS keys (Check whether every DNS key is valid)
- Verify Certificates (Check whether every certificate is valid)
- Federated Organization length (at least 2.5.9): Increase the max length of the organization field when adding a new server connection.
- Convert key label to text (at least 2.3.16): Some of the key labels that were created by MISP credit the loading of the data. This function will change them to "text".
- Fix duplicate attributes (at least 2.3.16): It was previously possible to add duplicate attributes (UDOs) in the database. This script will remove all duplicates and ensure that duplicates will not be entered into the database in the future.
- Rename duplicate events (at least 2.3.16): In some rare situations it could occur that a duplicate of an event was created on an instance, with the exact same UDO. The action will remove any such duplicate and make sure that the event happens only once.
- Purge orphaned attributes: In some rare occasions it can happen that you end up with some attributes in your database that do not belong to an event - for example during a race condition between an event insert and a delete. This can be checked and there are some orphaned attributes. If you have trouble in an issue where you cannot sort an attribute with a specific value, this is probably the reason.
- Clean regex table of potentially malicious entries (at least 2.3.16): More in the version it was possible to insert a regular expression that could be used to evaluate arbitrary code.
- Manage type attribute mapping (at least 2.3.17): New type UDO can cause issues with the MISP imports and so all these attributes will be modeled on entry to connect this. To connect existing entries, run this script.
- Index tables (This script will make indexes for all of the tables in MISP other than primary keys)
- Fix non-empty sharing group desc (This script will fix the sharing group desc in all non-empty sharing group sets events and attributes)

Upgrading a 2.3 instance to 2.4

Make sure you have a backup of your database. Make sure you test your installation before running these.

1. Upgrade to 2.4 - run this to export the 2.3 database to the 2.4 format.
2. If the previous succeeded, run the 2.3-to-2.4 script as script to remove the fields that are specific to 2.3. Make sure that the migration of the data to the 2.4 format was successful (use one script the result in the audit log). If you have run the 2.4 upgrade script previously but are running into SQL errors on the column new, run the script
**Server Settings**

Since version 2.3, MISP has a settings and diagnostics tool that allows site-admins to manage and diagnose their MISP installation. You can access this by navigating to Administration - Server settings.

**Server settings and diagnostics**

<table>
<thead>
<tr>
<th>Test</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall health</td>
<td>Critical, your MISP instance requires immediate attention. The overall health of your instance depends on the most severe unresolved issues.</td>
<td></td>
</tr>
<tr>
<td>Critical settings incorrectly set</td>
<td>Incorrect settings</td>
<td>MISP will not operate correctly or will be unresponsive until these issues are resolved.</td>
</tr>
<tr>
<td>Recommended settings incorrectly or not set</td>
<td>Incorrect settings</td>
<td>Some of the features of MISP cannot function until these issues are resolved.</td>
</tr>
<tr>
<td>Optional settings incorrectly or not set</td>
<td>Incorrect settings</td>
<td>There are some optional features that could be enabled to improve the health of your MISP instance.</td>
</tr>
<tr>
<td>Critical issues revealed by the diagnostics</td>
<td>I issues detected</td>
<td>Issues revealed here can be due to incorrect directory permissions or not correctly installed dependencies.</td>
</tr>
</tbody>
</table>
The settings and diagnostics tool is split up into several aspects, all accessible via the tabs on top of the tool. For any unset or incorrectly set setting, or failed diagnostic a number next to the tab name will indicate the number and severity of the issues. If the number is written with a red font, it means that the issue is critical. First, let’s look at the various tabs:

- **Overview**: General overview of the current state of your MISP installation
- **MISP settings**: Basic MISP settings. This includes the way MISP handles the default settings for distribution settings, whether background jobs are enabled, etc
- **GnuPG settings**: GnuPG related settings.
- **Proxy settings**: HTTP proxy related settings.
- **Security settings**: Settings controlling brute-force protection and the application’s salt key.
- **Misc settings**: Settings controlling debug options, please ensure that debug is always disabled on a production system.
- **Diagnostics**: The diagnostics tool checks if all directories that MISP uses to store data are writeable by the apache user. Also, the tool checks whether the STIX libraries and GnuPG are working as intended.
- **Workers**: Shows the background workers (if enabled) and shows a warning if they are not running. Admins can also restart the workers here.
- **Download report**: Download a report of all the settings visible in the tool, in JSON format.
Each of the setting pages is a table with each row representing a setting. Coloured rows indicate that the setting is incorrect / not set and the colour determines the severity (red = critical, yellow = recommended, green = optional). The columns are as follows:

- **Priority**: The severity of the setting.
- **Setting**: The setting name.
- **Value**: The current value of the setting.
- **Description**: A description of what the setting does.
- **Error Message**: If the setting is incorrect / not set, this field will let the user know what is wrong.

### Server settings

<table>
<thead>
<tr>
<th>Worker</th>
<th>Worker Id</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>cache</td>
<td>nginx:virtualBox034:cache</td>
<td>OK</td>
</tr>
<tr>
<td>default</td>
<td>nginx:virtualBox031:default</td>
<td>OK</td>
</tr>
<tr>
<td>email</td>
<td>nginx:virtualBox031:email</td>
<td><strong>OK</strong></td>
</tr>
</tbody>
</table>

**Restart all workers**  
This will restart all of the workers and refresh the page. Keep in mind, this process can take a few seconds to complete, so refresh the page again in 5-10 seconds to see the correct results.
The workers tab shows a list of the workers that MISP can use. You can restart workers using the "restart all workers" button. If the button doesn't work, make sure that the workers were started using the apache user. This can however only be done using the command line, refer to the INSTALL.txt documentation on how to let the workers automatically start on each boot.

- **Worker Type**: The worker type is determined by the queue it monitors. MISP currently has 6 queues (cache, default, prio, email, update and a special schdlr queue).
- **Worker Id**: The ID is made up of the machine name, the PID of the worker and the queue it monitors.
- **Status**: Displays OK if the worker is running. If the schdlr worker is the only one not running, make sure that you copy the config file into the cakeresque directory as described in the INSTALL.txt documentation.

### Worker types

- **cache**
  Role: Interdependence:

- **default**
  Role: Interdependence:

- **email**
  Role: Interdependence:

- **update**
  Role: Interdependence:

- **prio**
  Role: Interdependence:

- **scheduler**
  Role: Interdependence:

### Workers dead

Even if the workers are dead, any actions related to them are on-hold. Nothing is lost. Simply restarting the worker will resume any operations.

You can either relaunch them via the UI or manually by running `sudo -u www-data bash /var/www/MISP/app/Console/worker/start.sh` on the CLI. For reference, below is the script in question.

```bash
#!/usr/bin/env bash

# Check if run as root
if [ "$EUID" -eq 0 ]; then
  echo "Please DO NOT run the worker script as root"
  exit 1
fi

# Extract base directory where this script is and cd into it
cd "${0%/*}"
..../cake CakeResque.CakeResque stop --all
..../cake CakeResque.CakeResque start --interval 5 --queue default
..../cake CakeResque.CakeResque start --interval 5 --queue prio
..../cake CakeResque.CakeResque start --interval 5 --queue cache
..../cake CakeResque.CakeResque start --interval 5 --queue email
..../cake CakeResque.CakeResque startscheduler --interval 5

exit 0
```
Import Blacklist

It is possible to ban certain values from ever being entered into the system via an event info field or an attribute value. This is done by blacklisting the value in this section.

Adding and modifying entries

Administrators can add, edit or delete blacklisted items by using the appropriate functions in the list's action menu and the menu on the left.

Import Regexp

The system allows administrators to set up rules for regular expressions that will automatically alter newly entered or imported events (from GFI Sandbox).

The purpose of Import Regexp entries

They can be used for several things, such as unifying the capitalisation of file paths for more accurate event correlation or to automatically censor the usernames and use system path variable names (changing C:\Users\UserName\Appdata\Roaming\file.exe to %APPDATA%\file.exe). The second use is blocking, if a regular expression is entered with a blank replacement, any event info or attribute value containing the expression will not be added. Please make sure the entered regexp expression follows the `preg_replace` pattern rules as described here.

Adding and modifying entries

Administrators can add, edit or delete regular expression rules, these “expressions” are made up of a regex pattern that the system searches for and a replacement for the detected pattern.

<table>
<thead>
<tr>
<th>Id</th>
<th>Regex</th>
<th>Replacement</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>../ProgramData/</td>
<td>%ALLUSERSPROFILE\</td>
<td>ALL</td>
</tr>
<tr>
<td>2</td>
<td>../Documents and Settings.All Users/</td>
<td>%ALLUSERSPROFILE\</td>
<td>ALL</td>
</tr>
<tr>
<td>3</td>
<td>../Program Files.Common Files/</td>
<td>%COMMONPROGRAMFILES\</td>
<td>ALL</td>
</tr>
<tr>
<td>4</td>
<td>../Program Files (x86).Common Files/</td>
<td>%COMMONPROGRAMFILES(x86)\</td>
<td>ALL</td>
</tr>
<tr>
<td>5</td>
<td>../Users/(w).AppData.Local.Tempo/</td>
<td>%TEMP\</td>
<td>ALL</td>
</tr>
<tr>
<td>6</td>
<td>../ProgramData/</td>
<td>%PROGRAMDATA\</td>
<td>ALL</td>
</tr>
<tr>
<td>7</td>
<td>../Program Files/</td>
<td>%PROGRAMFILES\</td>
<td>ALL</td>
</tr>
</tbody>
</table>
Managing the Signature whitelist

The signature whitelist view, accessible through the administration menu on the left, allows administrators to create and maintain a list of addresses that are whitelisted from ever being added to the NIDS signatures. Addresses listed here will be commented out when exporting the NIDS list.

Whitelisting an address

While in the whitelist view, click on New Whitelist on the left to bring up the "add whitelist" view to add a new address.

Managing the list

When viewing the list of whitelisted addresses, the following data is shown: The ID of the whitelist entry (assigned automatically when a new address is added), the address itself that is being whitelisted and a set of controls allowing you to delete the entry or edit the address.

Import Whitelist

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><a href="http://www.futuremark.com">www.futuremark.com</a></td>
<td></td>
</tr>
</tbody>
</table>
Using MISP logs

Users with audit permissions are able to browse or search logs that MISP automatically appends each time certain actions are taken (actions that modify data or if a user logs in and out). Generally, the following actions are logged:

- **User**: Creation, deletion, modification, Login / Logout
- **Event**: Creation, deletion, modification, publishing
- **Attribute**: Creation, deletion, modification
- **ShadowAttribute**: Creation, deletion, Accept, Discard
- **Roles**: Creation, deletion, modification
- **Blacklist**: Creation, deletion, modification
- **Whitelist**: Creation, deletion, modification
- **Regexp**: Creation, deletion, modification

Browsing logs

Listing all the log entries will display the following columns generated by the users of your organisation (or all organisations in the case of site admins):

Logs
- **Id**: The automatically assigned ID number of the entry.
- **Email**: The e-mail address of the user whose actions triggered the entry.
- **Org**: The organisation of the above mentioned user.
- **Created**: The date and time when the entry originated.
- **Action**: The action's type. This can include: login/logout for users, add, edit, delete for events, attributes, users and servers.
- **Title**: The title of an event always includes the target type (Event, User, Attribute, Server), the target's ID and the target's name (for example: e-mail address for users, event description for events).
- **Change**: This field is only populated for entries with "add" or "edit" actions. The changes are detailed in the following format: `variable (initial_value) => (new_value)`,... When the entry is about the creation of a new item (such as adding a new event) then the change will look like this for example: `org() => (ADMIN), date() => (20012-10-19),...`

### Search Logs

<table>
<thead>
<tr>
<th>Email</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Action**

- ALL

<table>
<thead>
<tr>
<th>Title</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Searching Logs

Another way to browse the logs is to search it by filtering the results according to the following fields (the search is a sub-string search, the sub-string has to be an exact match for the entry in the field that is being searched for):

- **Email**: By searching by Email, it is possible to view the log entries of a single user.
- **Org**: Searching for an organisation allows you to see all actions taken by any member of the organisation.
- **Action**: With the help of this drop down menu, you can search for various types of actions taken (such as logins, deletions, etc).
- **Title**: There are several ways in which to use this field, since the title fields contain several bits of information and the search searches for any substrings contained within the field, it is possible to just search for the ID number of a logged event, the username / server's name / event's name / attributes name of the event target.
- **Change**: With the help of this field, you can search for various specific changes or changes to certain variables (Ex: using “Published” as the search term and find all log entries where an event has been "Published", ip-src will find all attributes where a source IP address has been entered / edited, etc).

Background Processing

If enabled, MISP can delegate a lot of the time intensive tasks to the background workers. These will then be executed in sequence, allowing the users of the instance to keep using the system without a hiccup and without having to wait for the process to finish. It also allows for certain tasks to be scheduled and automated.

Command Line Tools for the Background Workers

The background workers are powered by CakeResque, so all of the CakeResque commands work. To start all of the workers needed by MISP go to your `/var/www/MISP/app/Console/worker` (assuming a standard installation path) and execute `start.sh`. To interact with the workers, here is a list of useful commands. Go to your `/var/www/MISP/app/Console` (assuming a standard installation path) and execute one of the following commands as a parameter to `.cake CakeResque` (for example: `.cake CakeResque tail`):

- **start**: Start a new worker.
- **startscheduler**: Start a new scheduler worker.
- **stop**: Stop a worker.
- **pause**: Pause a worker.
- **resume**: Resume a paused worker.
- **cleanup**: Terminate the job that a worker is working on with immediate effect. You will be presented with a choice of workers to choose from when executing this command.
- **restart**: Stop all Resque workers, and start a new one.
- **clear**: Clear all jobs inside a queue
- **reset**: Reset CakeResque internal worker's saved status
- **stats**: Display some statistics about your workers including the count of successful and failed jobs.
- **tail**: Tail the various (workers) log files that CakeResque creates, just choose the one from the list that you are interested in.
- **track**: Track a job status.
- **load**: Load a set of predefined workers.

The other commands should not be required, instead of starting / stopping or restarting workers use the supplied start.sh (it stops all workers and starts them all up again). For further instructions on how to use the console commands for the workers, visit the CakeResque list of commands.

Monitoring the Background Processes

The "Jobs" menu item within the Administration menu allows site admins to get an overview of all of the current and past
The "Jobs" menu item within the Administration menu allows site admins to get an overview of all of the current and past scheduled jobs. Admins can see the status of each job, and what the queued job is trying to do. If a job fails, it will try to set an error message here too. The following columns are shown in the jobs table:

- **Id:** The job’s ID (this is the ID of the job’s metadata stored in the default datastore, not to be confused with the process ID stored in the redis database and used by the workers)
- **Process:** The process’s ID.
- **Worker:** The name of the worker queue. There are 3+1 workers running if background jobs are enabled: default, cache, email, and a special Scheduler (this should never show up in the jobs table).
- **Job Type:** The name of the queued job.
- **Input:** Shows a basic input handled by the job - such as "Event:50" for a publish email alert job for event 50.
- **Message:** This will show what the job is currently doing or alternatively an error message describing why a job failed.
- **Org:** The string identifier of the organisation that has scheduled the job.
- **Status:** The status reported by the worker.
- **Retries:** Currently unused, it is planned to introduced automatic delayed retries for the background processing and thus add resilience.
- **Progress:** A progress bar showing how the job is coming along.

### Jobs

<table>
<thead>
<tr>
<th>Id</th>
<th>Process</th>
<th>Worker</th>
<th>Job Type</th>
<th>Input</th>
<th>Message</th>
<th>Org</th>
<th>Status</th>
<th>Retries</th>
<th>Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>3999</td>
<td>e5f17abc2d0e8d417773d30b100</td>
<td>default</td>
<td>publish, event</td>
<td>Event:10</td>
<td>Event published</td>
<td>ADMIN</td>
<td>Completed</td>
<td>0</td>
<td>Completed</td>
</tr>
<tr>
<td>3992</td>
<td>17770d4b8c3b0b10710f14d025b6</td>
<td>default</td>
<td>publish, event</td>
<td>Event:10</td>
<td>Event published</td>
<td>ADMIN</td>
<td>Completed</td>
<td>0</td>
<td>Completed</td>
</tr>
<tr>
<td>3991</td>
<td>15d1774b2c8e9c0772c71577715</td>
<td>default</td>
<td>publish, event</td>
<td>Event:10</td>
<td>Event published</td>
<td>ADMIN</td>
<td>Completed</td>
<td>0</td>
<td>Completed</td>
</tr>
<tr>
<td>3990</td>
<td>828e5963e21f155763f717d8b100</td>
<td>default</td>
<td>publish, event</td>
<td>Event:10</td>
<td>Event published</td>
<td>ADMIN</td>
<td>Completed</td>
<td>0</td>
<td>Completed</td>
</tr>
<tr>
<td>3989</td>
<td>c87f17771d380a24a9554c24f8bd</td>
<td>default</td>
<td>publish, event</td>
<td>Event:10</td>
<td>Event published</td>
<td>ADMIN</td>
<td>Completed</td>
<td>0</td>
<td>Completed</td>
</tr>
</tbody>
</table>
Scheduling Jobs and Recurring Jobs

Apart from off-loading long-lasting jobs to the background workers, there is a second major benefit of enabling the background workers: Site-administrators can schedule recurring tasks for the jobs that generally take the longest to execute. At the moment this includes pushing / pulling other instances and generating a full export cache for every organisation and export type. MISP comes with these 3 tasks pre-defined, but further tasks are planned. The following fields make up the scheduled tasks table:

- **Id**: The ID of the task.
- **Type**: The type of the task.
- **Frequency (h)**: This number sets how often the job should be executed in hours. Setting this to 168 and picking the next execution on Sunday at 01:00 would execute the task every Sunday at 1 AM. Setting this value to 0 will make the task only run once on the scheduled date / time without rescheduling it afterwards.
- **Scheduled Time**: The time (in 24h format) when the task should be executed the next time it runs (and all consecutive times if a multiple of 24 is chosen for frequency).
- **Next Run**: The date on which the task should be executed.
- **Description**: A brief description of the task.
- **Message**: This field shows when the job was queued by the scheduler for execution.

### Scheduled Tasks

Here you can schedule pre-defined tasks that will be executed every X hours. You can alter the date and time of the next scheduled execution and the frequency at which it will be repeated (expressed in hours). If you set the frequency to 0 then the task will not be repeated. To change any of the above mentioned settings just click on the appropriate field and hit update all when you are done editing the scheduled tasks.

<table>
<thead>
<tr>
<th>Id</th>
<th>Type</th>
<th>Frequency (h)</th>
<th>Scheduled Time</th>
<th>Next Run</th>
<th>Description</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>push-all</td>
<td>0</td>
<td>12:00</td>
<td>2014-02-08</td>
<td>Initiates a full push for all eligible instances.</td>
<td>not scheduled yet</td>
</tr>
<tr>
<td>2</td>
<td>pull-all</td>
<td>0</td>
<td>12:00</td>
<td>2014-02-08</td>
<td>Initiates a full pull for all eligible instances.</td>
<td>not scheduled yet</td>
</tr>
<tr>
<td>1</td>
<td>cache export</td>
<td>36</td>
<td>18:00</td>
<td>2014-02-20</td>
<td>Generates export caches for every export type and for every organisation.</td>
<td>32 jobs started at 20/02/2014 - 17:07:27</td>
</tr>
</tbody>
</table>
Various administration tips & tricks

Setting a Publish Alert Filter

To regulate the reception of e-mail from MISP it is possible to create filters. Each individual user account can apply such filter. The filter can be configured by the user but also by the organization administrator.

After login goto Administration -> Set User Setting:
A new screen appears. Make sure the “Setting” drop down box shows “publish_alert_filter”:
The text field “Value” contains the filter, which needs to be provided in JSON format. Important JSON-objects which can be used here go by the name AND”, “OR” and “NOT”. These should be structured in a logical tree.

The filtering can be applied to tags or to a publishing organization.

In the following example, all notifications will be filtered which carry ‘tlp.white’ and ‘tlp.green’ in the name of the tag:

```
{
    "NOT": {
        "Tag.name": [ "tlp.white", "tlp.green" ]
    }
}
```

The publish_alert_filter setting allows one filter definition to be active.

After applying the configuration, the filter will show up in the “My Settings” menu:
Default sharing level

Choose your default sharing level to match your usage scenario for MISP. The setting is named `default_event_distribution` and the values can be:

- Your organisation only (default)
- This community only
- Connected communities
- All communities

You can also set a default distribution level for attributes contained in an event with `default_attribute_distribution`, and it has the same values as the default sharing level for events plus an additional one that allows attributes to inherit the sharing level of the event.

Adding organisation logos

You can add a logo for your organisations in MISP by uploading them via the tab Manage files under the Administration menu & Server Settings sub-menu. The filename must be exactly the same as the organisation name that you will use in MISP. It is recommended to use PNG files of 48x48 pixels.

The _schdlr_ worker is not starting

If you already made sure that you copied the config file under the cakeresque directory, it might be due to the FQDN of the server hosting the instance has changed. A way to fix this is to flush temporary data stored in redis. This can be done by logging in redis, for example when logging in with redis-cli, and issuing a flushall command.

How to redirect HTTP to HTTPS

Here is a sample configuration for Apache HTTPS.

```xml
<VirtualHost *:80>
    ServerAdmin misp@misp.misp
    ServerName misp.misp.misp
    ServerAlias misp-int.misp.misp

    Redirect permanent / https://misp.misp.misp

    LogLevel warn
    ErrorLog /var/log/apache2/misp.local_error.log
    CustomLog /var/log/apache2/misp.local_access.log combined
    ServerSignature Off

</VirtualHost>

<VirtualHost *:443>
    ServerAdmin misp@misp.misp
    ServerName misp.misp.misp
    ServerAlias misp-int.misp.misp

    DocumentRoot /var/www/MISP/app/webroot
    <Directory /var/www/MISP/app/webroot>
        Options -Indexes
        AllowOverride all
        Order allow,deny
        allow from all
    </Directory>

    SSLEngine On
    SSLCertificateFile /etc/ssl/misp.misp.misp/misp.crt
    SSLCertificateKeyFile /etc/ssl/misp.misp.misp/misp.key
</VirtualHost>
```
SSLCertificateChainFile /etc/ssl/misp.misp.misp/mispCA.crt
LogLevel warn
ErrorLog /var/log/apache2/misp.local_error.log
CustomLog /var/log/apache2/misp.local_access.log combined
ServerSignature Off
</VirtualHost>

Taken from Koen Van Impe's blog

**Increase max size of Samples / other files**

Trying to upload a large samples (>50M) might cause the following error: [!] 500 Server Error: Internal Server Error

Or will give you an error page in browser.

The error logs on the system will display the following:

```
PHP Warning: POST Content-Length of 57526024 bytes exceeds the limit of 8388608 bytes in Unknown on line 0, referer: https://XYZ/attributes/add_attachment/1948
```

And / Or

```
PHP Fatal error: Allowed memory size of 134217728 bytes exhausted (tried to allocate 76705009 bytes) in /var/www/MISP/app/Lib/cakephp/lib/Cake/Network/CakeRequest.php on line 996
```

To fix that you need to adjust the php settings:

```
vi /etc/php5/apache2/php.ini
```

Increase to the following values (or more if you want to)

```
; Maximum size of POST data that PHP will accept.
; Its value may be 0 to disable the limit. It is ignored if POST data reading
; is disabled through enable_post_data_reading.
; http://php.net/post-max-size
post_max_size = 256M
[-]
; Maximum amount of memory a script may consume (128MB)
; http://php.net/memory-limit
memory_limit = 1024M
```

And then restart apache2

```
service apache2 restart
```

**Support & feature requests**

The preferred method for support & feature requests is to use the GitHub ticketing system.

If you want to discuss something related to MISP, want some help from the community, etc… You have the MISP Users mailing list and the MISP developers mailing list.

A number of companies offer custom development, consulting, and support around MISP, please check the support page of the MISP Project website.
More information in the notification emails about new events

The setting MISP.extended_alert_subject allows you to have an extended subject. One word of warning though. If you’re using encryption: the subject will not be encrypted. Be aware that you might leak some sensitive information this way. Below is an example how the two subject types look like. First with the option disabled, then with the option enabled.

| Event 7 | Low - TLP Amber |
| Event 8 | OSINT - Dissecting XXX.. - Low - TLP Amber |

Taken from Koen Van Impe's blog

Get top API users

Enable the log_auth setting in the server settings. Optionally enable log_client_ip if you want to get stats per client ip. Log into your mysql server and run the following query:

```
select ip,email,count(id) as c from logs WHERE ip IS NOT NULL group by ip,email order by c desc limit 10;
```

This will give you a top 10 table per ip and username:

```
<table>
<thead>
<tr>
<th>ip</th>
<th>email</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2.3.4</td>
<td><a href="mailto:bob@nsa.gov">bob@nsa.gov</a></td>
<td>4124</td>
</tr>
<tr>
<td>5.6.7.8</td>
<td><a href="mailto:vladimir@kremlin.ru">vladimir@kremlin.ru</a></td>
<td>1932</td>
</tr>
<tr>
<td>9.10.11.12</td>
<td><a href="mailto:fred@somewhere.eu">fred@somewhere.eu</a></td>
<td>1317</td>
</tr>
<tr>
<td>13.14.15.16</td>
<td>SYSTEM</td>
<td>16</td>
</tr>
</tbody>
</table>
```

MISP Logs

By default, MISP has several layers of logs that can be used to trouble-shoot and monitor the system. Let's walk through each of the available logs:

- **Apache access logs**: Rotating logs generated by apache, logging each request, by default (on Ubuntu) they are found in /var/log/apache2/misp.local_access.log. The location can be changed via the apache conf file
- **Apache error logs**: Rotating logs generated by apache, logging error messages, by default (on Ubuntu) they are found in /var/log/apache2/misp.local_error.log. This error log file will generally not be used by MISP, however, if there is a PHP level error that prevents MISP from functioning you might have relevant entries here.
- **MISP error log**: Generated by MISP, logging any exceptions that occur during usage. These can be found in /var/www/MISP/app/tmp/logs/error.log (assuming default installation path). If you see errors in here and are stuck with an issue let us know via GitHub!
- **MISP debug log**: Generated by MISP, any debug messages and Notice level messages will be sent to this file. Generally less interesting, but can be helpful during debugging sessions. It should not be necessary to monitor this under normal usage. The file can be found in /var/www/MISP/app/tmp/logs/debug.log (assuming default installation path).
- **MISP worker error log**: Generated by MISP background workers, logging any exceptions generated during a background job. It is the equivalent of the MISP error log for background jobs, so if scheduled tasks, synchronisation or e-mailing with the workers enabled are causing issues, this is the place to check. It can normally be found at /var/www/MISP/app/tmp/logs/resque-worker-error.log
- **MISP worker logs**: Rotating logs generated by MISP background workers, logging any jobs executed by workers. This is part of the normal operation of background workers and doesn't have to be monitored, though it can help when debugging issues. Normally found at /var/www/MISP/app/tmp/logs/resque-[current date].log
- **MISP scheduler error log**: Generated by MISP scheduler worker, logging any exceptions generated during the scheduling
of a background job. It is the equivalent of the MISP error log for scheduled jobs. It can normally be found at /var/www/MISP/app/tmp/logs/resque-scheduler-error.log

- MISP scheduler logs: Rotating logs generated by MISP scheduler worker, logging any scheduling of jobs to be executed by workers. This is part of the normal operation of the scheduler worker and doesn't have to be monitored, though it can help when debugging issues. Normally found at /var/www/MISP/app/tmp/logs/resque-scheduler-[current date].log

### Logging of failed authentication attempts

By default, MISP logs all failed login and authentication attempts in the built in Audit logs. To view any such failed attempts, simply log in as a site admin and navigate to Audit - List logs.

There are two types of entries that will be interesting if you are looking for failed authentication attempts, entries of action "auth_fail" (for failed attempts to authenticate via the API key or the external authentication system) and login_fail (for failed login attempts via the login page). You can also search for any such entries using the Search Logs feature, simply choose the desired action from the two listed above and hit search.

What is logged:

<table>
<thead>
<tr>
<th>Auth method</th>
<th>Action</th>
<th>Failed credentials logged</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Webform</td>
<td>login_fail</td>
<td>None</td>
<td>Optional</td>
</tr>
<tr>
<td>API</td>
<td>auth_fail</td>
<td>API key</td>
<td>Optional</td>
</tr>
<tr>
<td>Webform</td>
<td>auth_fail</td>
<td>External auth key</td>
<td>Optional</td>
</tr>
</tbody>
</table>

In order to enable IP logging for any logged request in MISP, navigate to Administration - Server settings - MISP settings and enable the MISP.log_client_ip setting.

It is also possible to enable full logging of API and external authentication requests using the MISP.log_auth setting in the same location, but keep in mind that this is highly verbose and will log every request made. In addition to the information above, all accessed resource URLs are also logged.

### Clearing expired sessions

By default the garbage collection of sessions is disabled in PHP. It is possible to enable it, but it's not recommended and as such MISP provides a manual way of clearing the sessions.

Navigate to the diagnostics screen of MISP (Administration - Server settings - Diagnostics) and near the bottom of the page there will be a counter showing the count of currently stored expired sessions. Simply purge them by clicking the applicable button when the number grows too large.

### Troubleshooting MISP not connecting to redis but redis-cli working

If you have an IPv6 enabled OS, but an older redis version that does not support IPv6 (<v2.8), MISP might fail to connect to the redis server while redis-cli is working. The reason is that redis-cli is connecting to 127.0.0.1 directly, while the calls inside the CakeResque library used by MISP are done using "localhost" which resolves both to the IPv4 and IPv6 loopback addresses. For some reasons, the use of the IPv6 address is attempted first which fails.

You can confirm this by trying to connect to redis using `telnet localhost 6379`. If it fails, the error message should mention the IPv6 loopback address (::1).

Two ways to fix it:

1) Upgrade your redis to a server that supports IPv6 (v2.8+). This is the preferred recommendation.
2) Comment the localhost mapping to IPv6 address in /etc/hosts

**Errors about fields or tables**

If you have errors with fields or tables that you can see in the error.log or in the page (if you enabled debug or site_admin_debug settings), an easy fix to make most of them go away is to use the **clean cache** feature on the server settings menu, diagnostics tab.

An example of error message:

```
Error: [PDOException] SQLSTATE[42S22]: Column not found: 1054 Unknown column 'Task.job_id' in 'field list'
```

**Jobs**

The Jobs tab gives you an overview on any currently running jobs or jobs that were previously completed and their status.
Typically this is one of the places you would turn to even some background process might not complete as expected to get an indication on any issues related to user initiated Jobs.

For ease of use, you can filter the Jobs by 'All', 'Default', 'Email', 'Cache'

**TODO: Explain differences Default, Email, Cache**

You can also purge the entries, either only by completed status or purge all. This is not automated and needs to be done manually.

**Scheduled Tasks**

Straight from the UI:

"""Here you can schedule pre-defined tasks that will be executed every x hours. You can alter the date and time of the next scheduled execution and the frequency at which it will be repeated (expressed in hours). If you set the frequency to 0 then the task will not be repeated. To change and of the above mentioned settings just click on the appropriate field and hit update all when you are done editing the scheduled tasks.

Warning: Scheduled tasks come with a lot of caveats and little in regards of customisations / granularity. You can instead simply create cron jobs out of the console commands as described here: Automating certain console tasks """

The task scheduler is a sub-par component to enable minimal functionality in terms of automating certain MISP tasks. If you have a dedicated and conscious MISP Site Admin she can keep an eye on the Scheduler to make sure everything runs smoothly.

For better performance please use a real scheduler like your systems' crontab. As a rule of thumb: If you can click on it, MISP can automate it.

**MISP Backup**

Currently there exists this backup script simply called misp-backup.sh

All you need is to copy the the sample config and make sure it is correct. Then launch the script.

```
cd /var/www/MISP/tools/misp-backup
sudo -u www-data cp misp-backup.conf.sample misp-backup.conf
sudo ./misp-backup.sh
```

Script output:

```
/var/www/MISP/tools/misp-backup 2.4 ● $ sudo ./misp-backup.sh
File ./misp-backup.conf exists.
copy of org images and other custom images
MySQL Dump
/var/www/MISP/tools/misp-backup
MISP Backup Completed, OutputDir: /opt/backup
FileName: MISP-Backup-20181128_163215.tar.gz
FullName: /opt/backup/MISP-Backup-20181128_163214.tar.gz
```

**MISP Restore**

In a similar fashion you can restore your MISP instance with the misp-restore.sh script. Read the script for details.
Feeds

Feeds are remote or local resources containing indicators that can be automatically imported in MISP at regular intervals. Feeds can be structured in MISP format, CSV format or even free-text format. You can easily import any remote or local URL to store them in your MISP instance. It's a simple way to gather many external sources of information without any programming skills into MISP.

Feeds description can be also easily shared among different MISP instances as you can export a feed description as JSON and import it back in another MISP instance.

Managing feeds

[warning] A site admin role is required to perform these actions.

To do so, you first need to access the list of feeds, using the top menu.
Adding feeds

Then select the add feed option on the side menu.
Here you will have access to a dynamic form. Let's check each field by order.

Add MISP Feed

Add a new MISP feed source.

- Enabled

Name

Feed name

Provider

Name of the content provider

Input Source

Local

- Remove input after ingestion

Url

URL of the feed

Source Format

MISP Feed

Distribution

All communities

Default Tag

None

Filter rules:

Modify

Add
- **Enabled**: Is the feed active or not
- **Lookup Visible**: If this is not checked, the correlation will only show up to you, if checked, correlations are visible for other users as well
- **Name**: Just a name to identify the feed
- **Provider**: Name of the content provider
- **Input Source**: Where does the input come from
- Network: hosted somewhere outside the platform
- Local: Hosted on the local server. On this case, a new checkbox "Remove input after ingestion" will appear. If checked, the source is deleted after usage.

```
Input Source
Local

Remove input after ingestion
```
- URL: URL of the feed, where it is located (for Local hosted files, point to the manifest.json e.g. /home/user/feed-generator/output/manifest.json)

- The Source Format can be:
MISP Feed: The source points to a list of json formatted like MISP events.
Example: https://www.circl.lu/doc/misp/feed-osint

Freetext Parsed Feed:

Source Format
Freetext Parsed Feed

Target Event
New Event Each Pull

Target Event ID
Leave blank unless you want to reuse an existing event.

Exclusion Regex
Regex pattern, for example: "^https://myfeedurl/"

☐ Auto Publish
☐ Override IDS Flag
☐ Delta Merge
- Target Event: Which will be the event getting updated with the data from the feed. Can be either "New Event Each Pull" (A new event will be created each time the feed is pulled) or "Fixed Event" (A unique event will be updated with the new data. This event is determined by the next field)
Managing Feeds

- Target Event ID: The id of the event where the data will be added (if not set, the field will be set the first time the feed is fetched)
- Exclusion Regex: Add a regex pattern for detecting iocs that should be skipped (this can be useful to exclude any references to the actual report/feed for example)
- Auto Publish: If checked, events created thanks to the feed will be automatically published
- Override IDS Flag: If checked, the IDS flag will be set to false
- Delta Merge: If checked, only data coming from the last fetch are kept, the old ones are deleted.

Simple CSV Parsed Feed:

Source Format

| Simple CSV Parsed Feed |

Target Event

New Event Each Pull

Target Event ID

Leave blank unless you want to reuse an existing event.

Value field(s) in the CSV

2,3,4 (column position separated by commas)

Delimiter

, ,

Exclusion Regex

Regex pattern, for example: "^https://myfeedurl/"

☐ Auto Publish
☐ Override IDS Flag
☐ Delta Merge
- Target Event: Which will be the event getting updated with the data from the feed. Can be either "New Event Each Pull" (A new event will be created each time the feed is pulled) or "Fixed Event" (A unique event will be updated with the new data. This event is determined by the next field)
- Target Event ID: The id of the event where the data will be added (if not set, the field will be set the first time the feed is fetched)
- Exclusion Regex: Add a regex pattern for detecting iocs that should be skipped (this can be useful to exclude any references to the actual report/feed for example)
- Value field(s) in the CSV: Select one or several fields that should be parsed by the CSV parser and converted into MISP attributes
- Delimiter: Set the default CSV delimiter (default = ",")
- Auto Publish: If checked, events created thanks to the feed will be automatically published
- Override IDS Flag: If checked, the IDS flag will be set to false
- Delta Merge: If checked, only data coming from the last fetch are kept, the old ones are deleted.
- Distribution: Define the distribution option that will be set on the event created by the feed
- Default Tag: A default tag can be added to the created event(s)
- Filter rules: Here you can define which tags or organisations are allowed or blocked.
To add a tag (resp. organisation), first type it into the top middle (resp. bottom middle) text field. Then use the arrows that point to the outside to add it to the allowed or blocked tags (resp. organisations) list.
### Set pull rules

<table>
<thead>
<tr>
<th>Allowed Tags</th>
<th>Blocked Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>lip:white</td>
<td>lip:green</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Allowed Organisations</th>
<th>Blocked Organisations</th>
</tr>
</thead>
</table>

[Update] [Cancel]
To remove a tag (resp. organisation), select it in the list and click on the arrow pointing to the inside.
Feed correlation

If an indicator from an feed matches an indicator within a MISP event, it will show up as "Feed hits" in the event overview. The correlation will not show up in the correlation graph of the event.
Updating Python Dependencies

MISP requires a couple of python libraries to be installed for the entire set of functionalities to work properly.

These functionalities include for instance the different import and export tools, the binaries extraction from attachments or PyMISP.

Installation

We started using virtual environments in MISP to make the installation and maintenance of the python versions easier.

Either using the installation script to setup a running MISP on your machine, or starting using the automatically generated virtual machine will give you access to the latest version of the required python libraries installed within a virtual environment called virtualenv.

But if you are using an older MISP version, you may want to install the virtual environment

Set the virtual environment up

```bash
# Create a python3 virtualenv

# Make pip happy
sudo mkdir /var/www/.cache
sudo chown www-data:www-data /var/www/.cache
```

If you already have a venv directory, you can skip this step

Updating MISP and its dependencies

Keeping MISP up-to-date as much as possible is the safest way to avoid most of the potential issues.

It can be done either by using the Update button in the diagnostic tool available with the MISP UI, or by using the command line.

Updating MISP core

In order to update MISP dependencies, we first want to pull the latest MISP version, so we have the latest submodule references as well.
MISP version

Every version of MISP includes a json file with the current version. This is checked against the latest tag on github, if there is a version mismatch the tool will warn you about it. Make sure that you update MISP regularly.

Currently installed version… v2.4.117 (b825e44a3988db8430089b0ebf787af272e0588)
Latest available version… v2.4.117 (b825e44a3988db8430089b0ebf787af272e0588)
Status... OK

Update MISP  Update Progress
Once we have the latest MISP update, we can start updating the python libraries.

**Updating the python dependencies**

MISP is provided with a lot of submodules used to ensure all the additional functionalities work as expected. Thus it is important to keep those dependencies up-to-date.

```
sudo -H -u www-data git submodule update --init --recursive
```

**Updating python dependencies**

It is possible to check the status of all the python libraries required by MISP, using again the diagnostic tool.

**Advanced attachment handler**

The advanced attachment tools are used by the add attachment functionality to extract additional data about the uploaded sample.

<table>
<thead>
<tr>
<th>tool</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>pydeep</td>
<td>OK</td>
</tr>
<tr>
<td>lief</td>
<td>OK</td>
</tr>
<tr>
<td>magic</td>
<td>OK</td>
</tr>
<tr>
<td>pymisp</td>
<td>OK</td>
</tr>
</tbody>
</table>

**STIX and Cybox libraries**

Mitre's STIX and Cybox python libraries have to be installed in order for MISP's STIX export to work. Make sure that you install them (as described in the MISP installation instructions) if you receive an error below.

If you run into any issues here, make sure that both STIX and CyBox are installed as described in the INSTALL.bat file. The required versions are:

- **STIX**: >1.2.0.6
- **CyBox**: >2.1.0.18 dev0
- **mixbox**: 1.0.3
- **maec**: >4.1.0.14
- **STIX2**: >1.2.0
- **PyMISP**: >2.4.93

Other versions might work but are not tested/recommended.

**Yara**

This tool tests whether piyara, the library used by the yara export tool is installed or not.

<table>
<thead>
<tr>
<th>tool</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>piyara library installed</td>
<td>OK</td>
</tr>
</tbody>
</table>
If something is going wrong, updating the corresponding library will make the diagnostic happy.

```bash
# Update PyMISP
cd /var/www/MISP/PyMISP

# Update the advanced attachment handler libraries (PICK THE ONE.S YOU NEED TO UPDATE)
# pydeep
# lief
sudo -H -u www-data /var/www/MISP/venv/bin/pip install -U https://github.com/lief-project/packages/raw/master/latest/pylief-0.9.0.dev.zip
# python-magic
sudo -H -u www-data /var/www/MISP/venv/bin/pip install -U python-magic

# Update the STIX dependencies (PICK THE ONE.S YOU NEED TO UPDATE)
# STIX
cd /var/www/MISP/app/files/scripts/python-stix
# mixbox
cd /var/www/MISP/app/files/scripts/mixbox
# Cybox
cd /var/www/MISP/app/files/scripts/python-cybox
# MAEC
cd /var/www/MISP/app/files/scripts/python-maec
# STIX 2
cd /var/www/MISP/cti-python-stix2

# Update Yara python library
sudo -H -u www-data /var/www/MISP/venv/bin/pip install -U plyara
```

Note that if any of the STIX, Cybox, mixbox or MAEC update fails because of a `No such file or directory` error, you just have to `git clone` them and start again the `pip` command above.

```bash
cd /var/www/MISP/app/files/scripts
sudo -H -u www-data git clone https://github.com/CybOXProject/python-cybox.git
sudo -H -u www-data git clone https://github.com/STIXProject/python-stix.git
sudo -H -u www-data git clone https://github.com/CybOXProject/mixbox.git
```

If you want to use / update the ZeroMQ functionality, you can also install / update the zmq python library.

```bash
# Install zmq library
sudo -H -u www-data /var/www/MISP/venv/bin/pip install zmq
# Update zmq library
sudo -H -u www-data /var/www/MISP/venv/bin/pip install -U zmq
```

**Updating MISP modules**

Another set of dependencies you may want to update are **MISP modules**.

**MISP modules** have their own dependencies that need to be up-to-date as well as the modules scripts themselves.

Note that the following instructions consider your **MISP modules** are installed in the default path where we install them on our virtual machine or following the install script. Please change the path accordingly if needed.

```bash
# Change here the path if needed
```
You will then need to restart the modules, please refer to the documentation.
Automation and MISP API

- **Automation API**
  - **General**
    - Automation URL
    - Automation key
    - Accept and Content-Type headers
  - **Automation using PyMISP**
  - **Status Codes**
  - **Error Handling**
    - Wrong endpoint chosen ~ Example
  - **Search**
  - **Events management**
    - GET /events ~ Accepted Methods ~ Description
    - GET /events ~ Description ~ URL Arguments ~ Output ~ Example
    - POST /events ~ Example
    - DELETE /events ~ Description ~ URL Arguments ~ Output ~ Example
    - GET /events/index ~ Description ~ Output ~ Example
    - POST /events/addTag Add or remove tags from events
    - GET /events/pushEventToZMQ/ ~ Description ~ URL Arguments ~ Example
    - GET /events/nids NIDS rules export
    - GET /events/hids Hash - HIDS database export
    - GET /events/stix STIX export ~ Various ways to narrow down the search results of the STIX export
  - **Tag management**
    - POST /tags/add ~ Description
    - POST /tags/attachTagToObject ~ Description ~ URL Arguments ~ Response ~ Example
    - POST /tags/removeTagFromObject ~ Description ~ URL Arguments ~ Response ~ Example
    - GET /tags/tagStatistics/ ~ Description ~ Output ~ Example
  - **Attribute management**
    - POST /attributes/add/ ~ URL Arguments ~ Output ~ Example
    - GET /attributes ~ URL Arguments ~ URL Attributes ~ Output ~ Example
    - POST /attributes/delete/ ~ Description ~ URL Arguments ~ Output ~ Example
    - GET /attributes/attributeStatistics ~ Description ~ Output ~ Example
    - GET /attributes/describeTypes Describe types API ~ Example
  - **Server management**
    - GET /servers/getPyMISPVersion ~ Result ~ Example
    - GET /servers/getVersion ~ Result ~ Example
  - **Sightings**
    - POST /sightings/add/
  - **User management**
    - POST /admin/users/add
    - POST admin/users/edit/
    - POST admin/users/delete/ ~ Parameters ~ Example
    - GET admin/users ~ Description ~ Output ~ Example
    - GET admin/users/view/ ~ Description ~ Parameters ~ Output ~ Example
    - POST admin/users/add/
  - **Discussion API**
  - **Organisation management**
  - **Special Cases**
    - XML Export ~ JSON query format ~ XML query format ~ XML download and URL parameters
    - CSV export ~ Update 2.4.82
- RPZ export
- Text export
- RESTful searches with JSON result
  - POST /attributes/restSearch ~ Example
- RESTful searches with XML result export
- Export attributes of event with specified type as XML
- Filtering event metadata
- Download attachment or malware sample
- Download malware sample by hash
- Upload malware samples using the "Upload Sample" API
- Proposals API
- Sharing groups
- Enable, disable and fetching feeds via the API
- Sightings API
- Warninglists API
  - GET warninglists/index ~ Description ~ Parameters ~ Output ~ Example
  - GET warninglists/view/1 ~ Description ~ Parameters ~ Output ~ Example
- Attribute statistics API
- Additional statistics
- MISP modules
  - Description
    - GET /modules/ ~ Example ~ Output
    - POST /modules/queryEnrichment ~ Example

**Automation API**

Automation 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 NIDS signature generation for IP, domains, host names, user agents etc., and hash list generation for MD5/SHA1 values of file artefacts. Support for more attribute types is planned. To make this functionality available for automated tools an authentication key is used. This makes it easier for your tools to access the data without further form-based-authentication.

**General**

**Automation URL**

The documentation will include a default MISP URL in the examples. Don't forget to replace it with your MISP URL.

Default MISP URL in the documentation:

```
https://<misp url>/
```

**Automation key**

The authentication of the automation is performed via a secure key available in the MISP UI interface. Make sure you keep that key secret as it gives access to the entire database! The API key is available in the event actions menu under automation.

Since version 2.2 the usage of the authentication key in the URL is deprecated. Instead, pass the auth key in an Authorization header in the request. The legacy option of having the auth key in the URL is temporarily still supported but not recommended.
The authorization is performed by using the following header:

```
Authorization: YOUR API KEY
```

**Accept and Content-Type headers**

When performing your request, depending on the type of request, you might need to explicitly specify in what content type you want to get your results. This is done by setting one of the below Accept headers:

```
Accept: application/json
Accept: application/xml
```

When submitting data in a POST, PUT or DELETE operation you also need to specify in what content-type you encoded the payload. This is done by setting one of the below Content-Type headers:

```
Content-Type: application/json
Content-Type: application/xml
```

Example:

```
```

By appending .json or .xml the content type can also be set without the need for a header.

**Automation using 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. PyMISP is available including a documentation with various examples.

**Status Codes**

To be done

- 50x
- 400 - 499

**Error Handling**

**Wrong endpoint chosen**

Example

```
```
Search

It is possible to search in the database for a list of attributes or events based on a list of criterias.

To return attributes or events in a desired format, use the following URL and header settings:

URL:

YOUR_MISP_URL/attributes/restSearch
YOUR_MISP_URL/events/restSearch

Headers:

Accept: application/json
Content-type: application/json
Authorization: YOUR_API_KEY

The next feature to take care of then is the body of the query. This is where you are going to put your filters.

As an example, if we want to export all the IP addresses that have a TLP marking and not marked as TLP:red, you can find below the corresponding filters to use:

```
{
  "returnFormat": "json",
  "type": {
    "OR": [
      "ip-src",
      "ip-dst"
    ],
    "tags": {
      "NOT": [
        "tlp:red"
      ],
      "OR": [
        "tlp:%"
      ]
    }
  }
}
```

Find below a non exhaustive list of parameters that can be used to filter data in your search (some parameters specific to given export formats are not mentioned):

- **returnFormat**: Set the return format of the search (Currently supported: json, xml, openioc, suricata, snort - more formats are being moved to restSearch with the goal being that all searches happen through this API). Can be passed as the first parameter after restSearch or via the JSON payload.
- **limit**: Limit the number of results returned, depending on the scope (for example 10 attributes or 10 full events).
- **page**: If a limit is set, sets the page to be returned. page 3, limit 100 will return records 201->300).
- **value**: Search for the given value in the attributes' value field.
- **type**: The attribute type, any valid MISP attribute type is accepted.
- **category**: The attribute category, any valid MISP attribute category is accepted.
- **org**: Search by the creator organisation by supplying the organisation identifier.
- **tags**: To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'.
- **quickfilter**: Enabling this (by passing "1" as the argument) will make the search ignore all of the other arguments, except for
the auth key and value. MISP will return an xml / json (depending on the header sent) of all events that have a sub-string match on value in the event info, event orgc, or any of the attribute value1 / value2 fields, or in the attribute comment.

- **from**: Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event.
- **to**: Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event.
- **eventid**: The events that should be included / excluded from the search.
- **withAttachments**: If set, encodes the attachments / zipped malware samples as base64 in the data field within each attribute.
- **metadata**: Only the metadata (event, tags, relations) is returned, attributes and proposals are omitted.
- **uuid**: Restrict the results by uuid.
- **publish_timestamp**: Restrict the results by the timestamp of the last publishing of the event. The input can be a timestamp or a short-hand time description (7d or 24h for example). You can also pass a list with two values to set a time range (for example [“14d”, “7d”]).
- **last**: (Deprecated synonym for publish_timestamp) Restrict the results by the timestamp of the last publishing of the event. The input can be a timestamp or a short-hand time description (7d or 24h for example). You can also pass a list with two values to set a time range (for example [“14d”, “7d”]).
- **timestamp**: Restrict the results by the timestamp (last edit). Any event with a timestamp newer than the given timestamp will be returned. In case you are dealing with /attributes as scope, the attribute’s timestamp will be used for the lookup. The input can be a timestamp or a short-hand time description (7d or 24h for example). You can also pass a list with two values to set a time range (for example [“14d”, “7d”]).
- **published**: Set whether published or unpublished events should be returned. Do not set the parameter if you want both.
- **enforceWarninglist**: Remove any attributes from the result that would cause a hit on a warninglist entry.
- **to_ids**: By default (0) all attributes are returned that match the other filter parameters, irregardless of their to_ids setting. To restrict the returned data set to to_ids only attributes set this parameter to 1. You can only use the special "exclude" setting to only return attributes that have the to_ids flag disabled.
- **deleted**: If this parameter is set to 1, it will return soft-deleted attributes along with active ones. By using "only" as a parameter it will limit the returned data set to soft-deleted data only.
- **includeEventUuid**: Instead of just including the event ID, also include the event UUID in each of the attributes.
- **event_timestamp**: Only return attributes from events that have received a modification after the given timestamp. The input can be a timestamp or a short-hand time description (7d or 24h for example). You can also pass a list with two values to set a time range (for example [“14d”, “7d”]).
- **sgReferenceOnly**: If this flag is set, sharing group objects will not be included, instead only the sharing group ID is set.
- **eventinfo**: Filter on the event’s info field.
- **searchall**: Search for a full or a substring (delimited by % for substrings) in the event info, event tags, attribute tags, attribute values or attribute comment fields.
- **attackGalaxy**: Select the ATT&CK matrix like galaxy to use when using returnFormat = attack. Defaults to the Mitre ATT&CK library via mitre-attack-pattern.

---

**Events management**

/events

**Accepted Methods**

- GET
- POST
- PUT
- DELETE

**Description**
Receive, update or delete Events. There is also a good amount of special output formats that can be triggered.

GET /events

Description
Receive events based on criteria

URL Arguments
- event_id: Event id to receive
- event_uuid: Event uuid to receive

Output
```
[{
    "id":1,
    "org_id":1,
    "date":"2014-12-10",
    "info":"OSINT - F-Secure W32/Regin, Stage #1",
    "orgc_id":2,
    "timestamp":141827625,
    "distribution":3,
    "sharing_group_id":0,
    "proposal_email_lock":false,
    "locked":false,
    "threat_level_id":1,
    "publish_timestamp":1515749192,
    "disable_correlation":false,
    "Org":{
        "id":1,
        "name":"ORGNAME"
    },
    "Orgc":{
        "id":2,
        "name":"CIRCL"
    },
    "EventTag": [{
        "id":1,
        "event_id":1,
        "tag_id":1,
        "Tag":{
            "id":1,
            "name":"Type:OSINT",
            "colour="#1eed40",
            "exportable":true
        }
    }],
    "SharingGroup": null
}]
```

Example
```

Example
```
```

That is how an event JSON object should look like
```
{"Event": {
    "date": "2015-01-01",
    "threat_level_id": 1,
    "info": "testevent",
    "published": false,
    "distribution": 0,
    "Attribute": {
        "type": "domain",
        "category": "Network activity",
        "to_ids": false,
        "distribution": 0,
        "comment": "",
        "value": "test.com"
    }
}}
```

DELETE /events

Description
Delete events based on criteria

URL Arguments
- event_id: Event id to receive
- event_uuid: Event uuid to receive
Output

```
{
  "name": "Event deleted.",
  "message": "Event deleted.",
  "url": "\events\delete\1"
}
```

Example

```
```

GET /events/index

Description

Return the event index. - Warning, there's a limit on the number of results

Output

```

Example

```
curl --header "Authorization: YOUR API KEY " --header "Accept: application/json" --header "Content-Type: application/json" -X DELETE http://10.50.13.60/events/1
```

POST /events/addTag Add or remove tags from events

You can add or remove an existing tag from an event in the following way:

https://<misp url>/events/addTag
https://<misp url>/events/removeTag

Just POST a JSON object in the following format (to the appropriate API depending on whether you want to add or delete a tag from an event):

```
{"request": {"Event": {"id": "228", "tag": "8"}}}
```

Where "tag" is the ID of the tag. You can also use the name of the tag the following way (has to be an exact match):

```
{"request": {"Event": {"id": "228", "tag": "OSINT"}}}
```
GET /events/pushEventToZMQ/

Description
Will push an Event to ZMQ

URL Arguments
- event_id

Example


GET /events/nids NIDS rules export

Automatic export of all network related attributes is available under the Snort or Suricata rule format. Only published events and attributes marked as IDS Signature are exported.

You can configure your tools to automatically download the following file:

https://<misp url>/events/nids/suricata/download
https://<misp url>/events/nids/snort/download

The full API syntax is as follows:

https://<misp url>/events/nids/[format]/download/[eventid]/[frame]/[tags]/[from]/[to]/[last]

format
The export format, can be "suricata" or "snort"

eventid
Restrict the download to a single event

frame
Some commented out explanation framing the data. The reason to disable this would be if you would like to concatenate a list of exports from various select events in order to avoid unnecessary duplication of the comments.

tags
To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'. You can also chain several tag commands together with the '&&' operator. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead). For example, to include tag1 and tag2 but exclude tag3 you would use:

https://<misp url>/events/nids/snort/download/false/false/tag1&&tag2&&!tag3

from
Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event.

to
Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of
the event.

_last_
Events published within the last x amount of time, where x can be defined in days, hours, minutes (for example 6d or 12h or 30m). This filter will use the published timestamp of the event.

The keywords false or null should be used for optional empty parameters in the URL.

An example for a Suricata export for all events excluding those tagged tag1, without all of the commented information at the start of the file would look like this:

https://<misp url>/events/nids/suricata/download/null/true/!tag1

Administration is able to maintain a white-list containing host, domain name and IP numbers to exclude from the NIDS export.

**GET /events/hids Hash - HIDS database export**

Automatic export of MD5/SHA1 checksums contained in file-related attributes. This list can be used to feed forensic software when searching for suspicious files. Only published events and attributes marked as IDS Signature are exported.

You can configure your tools to automatically download all the MD5 hashes from MISP:

https://<misp url>/events/hids/md5/download

Or the SHA1 hashes:

https://<misp url>/events/hids/sha1/download

The API's full format is as follow:

https://<misp url>/events/hids/[format]/download/[tags]/[from]/[to]/[last]

**format**
The export format, can be "md5" or "sha1"

**tags**
To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'. You can also chain several tag commands together with the '&&' operator. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead). For example, to include tag1 and tag2 but exclude tag3 you would use:

https://<misp url>/events/hids/md5/download/tag1&&tag2&&!tag3

**from**
Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event.

**to**
Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event.

**last**
Events published within the last x amount of time, where x can be defined in days, hours, minutes (for example 5d or 12h or 30m). This filter will use the published timestamp of the event.
The keywords false or null should be used for optional empty parameters in the URL.

For example, to only show sha1 values from events tagged tag1, use:

```
https://<misp url>/events/hids/sha1/download/tag1
```

### GET /events/stix STIX export

You can export MISP events in MITRE’s STIX format (to read more about [STIX](#)). The STIX XML export is currently very slow and can lead to timeouts with larger events or collections of events. The STIX JSON return format does not suffer from this issue.

**Usage of the API:**

```
https://<misp url>/events/stix/download
```

Search parameters can be passed to the function via URL parameters or by POSTing an xml or json object (depending on the return type). The following parameters can be passed to the STIX export tool: id, withAttachments, tags. Both id and tags can use the && (and) and ! (not) operators to build queries. Using the URL parameters, the syntax is as follows:

```
https://<misp url>/events/stix/download/[id]/[withAttachments]/[tags]/[from]/[to]/[last]
```

- **id**
  - The event’s ID

- **withAttachments**
  - Encode attachments where applicable

- **tags**
  - To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'. You can also chain several tag commands together with the '&&' operator. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead).

  For example, to include tag1 and tag2 but exclude tag3 you would use:

  ```
  https://<misp url>/events/stix/download/false/true/tag1&&tag2&&!tag3
  ```

- **from**
  - Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event.

- **to**
  - Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event.

- **last**
  - Events published within the last x amount of time, where x can be defined in days, hours, minutes (for example 5d or 12h or 30m). This filter will use the published timestamp of the event.

You can post an XML or JSON object containing additional parameters in the following formats.

If you use JSON query objects:

```
https://<misp url>/events/stix/download.json
```
If you use XML query objects:

https://<misp url>/events/stix/download

Various ways to narrow down the search results of the STIX export

For example, to retrieve all events tagged "APT1" but excluding events tagged "OSINT" and excluding events #51 and #62 without any attachments:

https://<misp url>/events/stix/download/!51&&!62/false/APT1&&!OSINT/2015-02-15

To export the same events using a POST request use:

https://<misp url>/events/stix/download.json

Together with this JSON object in the POST message:

{"request": [{"id": ["!51", ",!62"], "tags": ["APT1", ",!OSINT"], "from": "false", "to": "2015-02-15"}]

XML is automatically assumed when using the STIX export:

https://<misp url>/events/stix/download

The same search could be accomplished using the following POSTed XML object (note that ampersands need to be escaped, or alternatively separate id and tag elements can be used):

<request>&id=!51&!62</id>&tags="APT1"&!OSINT"&from=false"&to="2015-02-15"/></request>

Tag management

POST /tags/add

Description

POST /tags/attachTagToObject

Description

Attaches a Tag to an Object by a given UUID

URL Arguments
- **tag**
- **UUID**

### Response

```
{
  "name": "Tag tlp3Awhite(7) successfully attached to Attribute(153).",
  "message": "Tag tlp3Awhite(7) successfully attached to Attribute(153).",
  "url": "/tags/attachTagToObject"
}
```

### Example

```
curl --header "Authorization: YOUR API KEY " --header "Accept: application/json" --header "Content-Type: application/json" -X POST http://10.50.13.60/tags/attachTagToObject/5a0d68b3-6da0-4ced-8233-77bb950d210f/tlp3Awhite
```

### POST /tags/removeTagFromObject

#### Description

Removes an Tag to an Object by a given UUID

#### URL Arguments

- **tag**
- **UUID**

#### Response

```
{
  "name": "Tag tlp3Awhite(7) successfully removed from Attribute(153).",
  "message": "Tag tlp3Awhite(7) successfully removed from Attribute(153).",
  "url": "/tags/removeTagFromObject"
}
```

### Example

```
curl --header "Authorization: YOUR API KEY " --header "Accept: application/json" --header "Content-Type: application/json" -X POST http://10.50.13.60/tags/removeTagFromObject/5a0d68b3-6da0-4ced-8233-77bb950d210f/tlp3Awhite
```

### GET /tags/tagStatistics/

#### Description

Will give an overview of the used attribute tags

#### Output
Example

```bash
```

Attribute management

**POST /attributes/add/**

Adds an Attribute to an event

**URL Arguments**

- event id

**Output**

Example

```bash
curl --header "Authorization: YOUR API KEY " --header "Accept: application/json" --header "Content-Type: application/json" -d "{"event_id":"3542","value":"1.2.3.4","category":"Network activity","type":"ip-dst"}" http://10.50.13.60/attributes/add/3542
```

GET /attributes

Get an attribute

**URL Arguments**

- attribute uuid

**URL Attributes**

**Output**

```json
{"Attribute":{"id":39,"event_id":"1","object_id":0,"object_relation":null,"category":"Payload installation","type":"md5","to_ids":true,"uuid":"548847db-060c-4275-a8c7-15bb950d210b","timestamp":"1418217435","distribution":3,"sharing_group_id":0,"comment":"Regen samples collected.","deleted":false,"disable_correlation":false,"value":"049436bb90f71cf38549817d0b90e7da","event_uuid":"54884656-2da8-4625-bf57-43ef950d210b"}}
```
Example


POST /attributes/delete/

Description
Delete attributes.

URL Arguments
- attribute uuid
- attribute id
- attribute id/1 <-- hard delete

Output
{"message":"Attribute deleted."}

Example


Hard delete:


GET /attributes/attributeStatistics

Description
Will give an overview of the used attribute types

Output
{
  "attachment": "1",
  "comment": "1",
  "filename": "2",
  "float": "2",
  "ip-dst": "90",
  "ip-dst|port": "3",
  "link": "3",
}
Example


GET /attributes/describeTypes Describe types API

MISP can procedurally describe all attribute types and attribute categories it currently supports including the category - type mappings. To access this information simply send a GET request to:

Example

https://<misp url>/attributes/describeTypes

Depending on the headers passed the returned data will be a JSON object or an XML, with 3 main sections: types, categories, category_type_mappings.

Server management

GET /servers/getPyMISPVersion

Result

{"version":"2.4.85"}

Example


GET /servers/getVersion

Result

{"version":"2.4.85","perm_sync":true}

Example

Sightings

**POST /sightings/add/**

- attribute_id
- attribute_uuid

The different sightings types are:

```
0 => 'sighting',
1 => 'false-positive',
2 => 'expiration'
```

User management

MISP allows administrators to create and manage users via its REST API

https://<misp url>/admin/users/view/[user id]

**POST /admin/users/add**

To create a new user, send a POST request to:

**Sample input**

```
{
    "email":"andras.iklody@circl.lu",
    "org\_id":1,
    "role\_id":1
}
```

To view the mandatory and optional fields, use a GET request on the above URL.

**Sample output**

```
{
    "name": "\//admin\/users\/add API description",
    "description": "POST a User object in JSON format to this API to create a new user.",
    "mandatory_fields": [
        "email",
        "org\_id",
        "role\_id"
    ],
    "optional_fields": [
        "password",
        "external_auth_required",
        "external_auth_key",
        "enable_password",
        "nids_sid",
        "server_id",
        "gpgkey",
        "certif\_public",
        "autoalert",
        "contactalert",
        "disabled",
        "change\_pw"
    ]
}```
POST admin/users/edit/

To edit an existing user send a POST request to:

https://<misp url>/admin/users/edit/[user id]

Only the fields POSTed will be updated, the rest is left intact. To view all possible parameters, simply send a GET request to the above URL.

POST admin/users/delete/

You can also delete users by POSTing to the below URL, but keep in mind that disabling users (by setting the disabled flag via an edit) is always preferred to keep user associations to events intact.

Parameters

- [user id]

Example

https://<misp url>/admin/users/delete/[user id]

GET admin/users

Description

Will output all users

Output

[  
  
  "User": { 
    "id": "1",
    "password": "FOOOOOOOOOO",
    "org_id": "1",
    "server_id": "0",
    "email": "admin@admin.test",
    "autoalert": false,
    "authkey": "YOUR_API_KEY",
    "invited_by": "0",
    "gpgkey": null,
    "certif_public": "",
    "nids_sid": "40000000",
    "termsaccepted": true,
    "newsread": "0",
    "role_id": "1",
    "change_pw": "0",
    "contactalert": false,
    "disabled": false,
  
  
]
Example


GET admin/users/view/

Description

Will return a single user. To view a user simply send a GET request.

Parameters

- id

Output

```json
{
   "User": {
      "id": "1",
      "password": "******",
      "org_id": "1",
      "server_id": "0",
      "email": "admin@admin.test",
      "autoalert": false,
      "authkey": "YOUR API KEY",
      "invited_by": "0",
      "gpgkey": null,
      "certif_public": "",
      "nids_sid": "400000000",
      "termsaccepted": true,
      "newsread": "0",
      "role_id": "1",
      "change_pw": "0",
      "contactalert": false,
      "disabled": false,
      "expiration": null,
      "current_login": "1515752313",
      "last_login": "1515748671",
      "force_logout": false,
      "orgAdmins": []
   }
}
```
Example

curl --header "Authorization: YOUR API KEY " --header "Accept: application/json" --header "Content-Type: application/json" -X GET http://10.50.13.60/admin/users/view/1

POST admin/users/add/

Discussion API

If you would like to fetch a discussion thread including all of its posts, simply send a GET request to:

https://<misp url>/threads/view/<thread id>

Using the following headers:

Authorization: [Your auth key]
Content-type: application/json
Accept: application/json

To get all posts related to an event simply send a GET request to:

https://<misp url>/threads/viewEvent/<event id>

Organisation management

MISP allows administrators to create and manage organisations via its REST API

The API is available in JSON format so make sure you use the following headers:

Authorization: [Your auth key]
Content-type: application/json
Accept: application/json

To fetch all organisations send a GET request to:

https://<misp url>/organisations

To view an individual organisation, send a get request to:

https://<misp url>/organisations/view/id

The management of users happens via three apis:

https://<misp url>/admin/organisations/add
https://<misp url>/admin/organisations/edit/[org id]
https://<misp url>/admin/organisations/delete/[org id]
To delete an organisation simply send a POST or DELETE request to the above URL.

For creating or modifying an organisation, simply POST a JSON containing the relevant fields to the appropriate API. The only mandatory field is the organisation name, with a host of optional parameters.

An example for a simple organisation object:

```json
{
    "name": "Blizzard",
    "nationality": "US"
}
```

Not setting a field will assume the default settings for the given field in the case of a new organisation whilst it would retain the existing setting for existing organisations. The above example would create the following object in MISP:

```json
{
    "Organisation": {
        "id": "1108",
        "name": "Blizzard",
        "alias": "",
        "anonymise": false,
        "date_created": "2017-01-22 17:32:29",
        "date_modified": "2017-01-22 17:32:29",
        "description": "",
        "type": "",
        "nationality": "US",
        "sector": "",
        "created_by": "1",
        "uuid": "b8b4de9d-04f0-4d7d-bf15-0b88c8a83805",
        "contacts": "",
        "local": true,
        "landingpage": ""
    }
}
```

To query the add or edit APIs for the valid parameters, simply send a GET requests to either API. The result currently looks like this (which might change when new fields are added):

```json
{
    "name": "\admin\organisations\add API description",
    "description": "POST an Organisation object in JSON format to this API to create a new organisation.",
    "mandatory_fields": [
        "name"
    ],
    "optional_fields": [
        "anonymise",
        "description",
        "type",
        "nationality",
        "sector",
        "uuid",
        "contacts",
        "local"
    ],
    "url": "\admin\organisations\add"
}
```

**Special Cases**

**XML Export**
An automatic export of all events and attributes (except file attachments) is available under a custom XML format.

You can configure your tools to automatically download the following file:

https://<misp url>/events/xml/download

If you only want to fetch a specific event append the eventid number:

https://<misp url>/events/xml/download/1

You can post an XML or JSON object containing additional parameters in the JSON query format or XML query format. Query parameters provide a way to filter the output to specific parameters.

**JSON query format**

The URL is appended with json:

https://<misp url>/events/xml/download.json

The query parameters can be the following:

```
{"request": {"eventid": ["!51", "!62"], "withAttachment": false, "tags": ["APT1", "!OSINT"], "from": false, "to": "2015-02-15"}}
```

**XML query format**

The URL is path is:

https://<misp url>/events/xml/download

The query parameters can be the following:

```
<request><eventId>!51</eventId><eventId>!62</eventId><withAttachment>false</withAttachment><tags>APT1</tags><tags>!OSINT</tags><from>false</from><to>2015-02-15</to></request>
```

**XML download and URL parameters**

The XML download also accepts two additional the following optional parameters in the url:

https://<misp url>/events/xml/download/[eventid]/[withattachments]/[tags]/[from]/[to]/[last]

**eventid**

Restrict the download to a single event

**withattachments**

A boolean field that determines whether attachments should be encoded and a second parameter that controls the eligible tags.

**tags**

To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'. You can also chain several tag commands together with the '&&' operator. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead). For example, to include tag1 and tag2 but exclude tag3 you would use:

```xml
<request><eventId>!51</eventId><eventId>!62</eventId><withAttachment>false</withAttachment><tags>tag1</tags><tags>tag2</tags><tags>!tag3</tags><from>false</from><to>2015-02-15</to></request>
```
from
Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event.

to
Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event.

last
Events published within the last x amount of time, where x can be defined in days, hours, minutes (for example 5d or 12h or 30m). This filter will use the published timestamp of the event.

The keywords false or null should be used for optional empty parameters in the URL. Also check out the User Guide to read about the REST API.

CSV export
An automatic export of attributes is available as CSV. Only attributes that are flagged "to_ids" will get exported.

You can configure your tools to automatically download the following file:

https://<misp url>/events/csv/download

This will download all the valid attributes in your MISP instance (might take some time).

You can also configure your tools to download the attributes from a specific event. Here is the old legacy CSV export that will work like exporting all attributes:

https://<misp url>/events/csv/download/<event-id>

You can specify additional flags for CSV exports as follows:

POST to:

https://<misp url>/events/csv/download

Headers:

Authorization: <your auth key>
Content-type: application/json

Body:

{"parameter1":"value1", "parameter2":1, "parameter3":[]}

**eventid**
Restrict the download to a single event

**ignore**
Setting this flag to true will include attributes that are not marked "to_ids".

**tags**
Simply add a list of tags that should be included or negated (by prepending the tag name with a "!"). Any event with a negated tag will be ignored, even if an included tag is matching. An example is included further down.

**category**
The attribute category, any valid MISP attribute category is accepted.

**type**
The attribute type, any valid MISP attribute type is accepted.

**includeContext**
Include the event data with each attribute.

**from**
Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event.

**to**
Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event.

**last**
Events published within the last x amount of time, where x can be defined in days, hours, minutes (for example 5d or 12h or 30m). This filter will use the published timestamp of the event.

For example, to only download a csv generated of the "domain" type and the "Network activity" category attributes all events except for the one and further restricting it to events that are tagged "tag1" or "tag2" but not "tag3", only allowing attributes that are IDS flagged use the following syntax:

POST to:

```plaintext
https://<misp url>/events/csv/download
```

Headers:

```plaintext
Authorization: <your auth key>
Content-type: application/json
```

Body:

```json
{
  "tags": ["tag1", "tag2", "!tag3"],
  "category": "Network activity",
  "type": "domain"
}
```

Alternatively you can fall back to the deprecated syntax of passing parameters in a GET request via the URL, however this is discouraged:

```plaintext
https://<misp url>/events/csv/download/[eventid]/[ignore]/[tags]/[category]/[type]/[includeContext]/[from]/[to]/[last]
```

If you use the deprecated URL parameter method, keep in mind that the keywords false or null should be used for optional empty parameters. To export the attributes of all events that are of the type "domain", use the following syntax:

```plaintext
https://<misp url>/events/csv/download/false/false/false/false/domain
```

**Update 2.4.82**

Since version 2.4.82, the new export format allows to select more columns using the following query format:
The order of columns will be honoured including those related to object level information.

To select object level columns, simply prepend the given object column’s name by object_, such as:

https://<misp-instance>/events/csv/download/<event-id>?attributes=timestamp,type,uuid,value&object_attributes=uid,name

The following columns will be returned (all columns related to objects will be prefixed with object_):

timestamp,type,uuid,value,object_uuid,object_name

The includeContext option includes the tags for the event for each line.

**RPZ export**

You can export RPZ zone files for DNS level firewall by using the RPZ export functionality of MISP. The file generated will include all of the IDS flagged domain, hostname and IP-src/IP-dst attribute values that you have access to.

It is possible to further restrict the exported values using the following filters:

- **tags**
  - To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'. You can also chain several tag commands together with the '&&' operator. Please be aware the colons (:) cannot be used in the tag search when passed through the URL. Use semicolons instead (the search will automatically search for colons instead).

- **id**
  - The event’s ID

- **from**
  - Events with the date set to a date after the one specified in the from field (format: 2015-02-03)

- **to**
  - Events with the date set to a date before the one specified in the to field (format: 2015-02-03)

MISP will inject header values into the zone file as well as define the action taken for each of the values that can all be overwritten. By default these values are either the default values shipped with the application, or ones that are overwritten by your site administrator. The values are as follows:

<table>
<thead>
<tr>
<th>Value name</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPZ_policy</td>
<td>DROP</td>
</tr>
<tr>
<td>RPZ_walled_garden</td>
<td>127.0.0.1</td>
</tr>
<tr>
<td>RPZ_serial</td>
<td>$date00</td>
</tr>
<tr>
<td>RPZ_refresh</td>
<td>2h</td>
</tr>
<tr>
<td>RPZ_retry</td>
<td>30m</td>
</tr>
<tr>
<td>RPZ_expiry</td>
<td>30d</td>
</tr>
<tr>
<td>RPZ_minimum_ttl</td>
<td>1h</td>
</tr>
<tr>
<td>RPZ_ttl</td>
<td>1w</td>
</tr>
<tr>
<td>RPZ_ns</td>
<td>localhost.</td>
</tr>
</tbody>
</table>
To override the above values, either use the URL parameters as described below:

```
https://<misp url>/attributes/rpz/download/[tags]/[eventId]/[from]/[to]/[policy]/[walled_garden]/[ns]/[ns_alt]/
[email]/[serial]/[refresh]/[retry]/[expiry]/[minimum_ttl]/[ttl]
```

Or POST an XML or JSON object with the above listed options:

```
<request>
  <tags>OSINT&&!OUTDATED</tags>
  <policy>walled-garden</policy>
  <walled_garden>teamliquid.net</walled_garden>
  <refresh>5h</refresh>
</request>
```

```
{"request": {"tags": ["OSINT", "!OUTDATED"], "policy": "walled-garden", "walled_garden": "teamliquid.net", "refresh": "5h"}}
```

### Text export

An export of all attributes of a specific type to a plain text file. By default only published and IDS flagged attributes are exported.

You can configure your tools to automatically download the following files:

```
https://<misp url>/attributes/text/download/md5
https://<misp url>/attributes/text/download/sha1
https://<misp url>/attributes/text/download/sha256
https://<misp url>/attributes/text/download/filename
https://<misp url>/attributes/text/download/filename|md5
https://<misp url>/attributes/text/download/filename|sha1
https://<misp url>/attributes/text/download/filename|sha256
https://<misp url>/attributes/text/download/ip-src
https://<misp url>/attributes/text/download/ip-dst
https://<misp url>/attributes/text/download/hostname
https://<misp url>/attributes/text/download/domain
https://<misp url>/attributes/text/download/email-src
https://<misp url>/attributes/text/download/email-dst
https://<misp url>/attributes/text/download/email-subject
https://<misp url>/attributes/text/download/email-attachment
https://<misp url>/attributes/text/download/url
https://<misp url>/attributes/text/download/user-agent
https://<misp url>/attributes/text/download/registry
https://<misp url>/attributes/text/download/registry|value
https://<misp url>/attributes/text/download/AS
https://<misp url>/attributes/text/download/snort
https://<misp url>/attributes/text/download/pattern-in-traffic
https://<misp url>/attributes/text/download/yara
https://<misp url>/attributes/text/download/vulnerability
https://<misp url>/attributes/text/download/attachment
https://<misp url>/attributes/text/download/malware-sample
https://<misp url>/attributes/text/download/link
https://<misp url>/attributes/text/download/comment
https://<misp url>/attributes/text/download/text
https://<misp url>/attributes/text/download/other
https://<misp url>/attributes/text/download/named_pipe
https://<misp url>/attributes/text/download/mutex
https://<misp url>/attributes/text/download/target-user
https://<misp url>/attributes/text/download/target-email
```
To restrict the results by tags, use the usual syntax. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead). To get ip-src values from events tagged tag1 but not tag2 use:

https://<misp url>/attributes/text/download/ip-src/tag1&&

It is possible to restrict the text exports on additional flags. The first allows the user to restrict based on event ID, whilst the second is a boolean switch allowing non IDS flagged attributes to be exported. Additionally, choosing "all" in the type field will return all eligible attributes.

https://<misp url>/attributes/text/download/[type]/[tags]/[event_id]/[allowNonIDS]/[from]/[to]/[last]

type

The attribute type, any valid MISP attribute type is accepted.

tags

To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'. You can also chain several tag commands together with the '&&' operator. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead).

allowNonIDS

Include attributes that would normally be excluded due to the IDS flag not being set or due to being whitelisted

from

Set the lowest "date" field value that should be included in the export (format YYYY-MM-DD)

to

Set the highest "date" field value that should be included in the export (format YYYY-MM-DD)

last

Set the timeframe of the export based on the "timestamp" value. The parameter uses a time + metric notation (valid examples: "2w", "60m", "24h")

For example, to include tag1 and tag2 but exclude tag3 you would use:

https://<misp url>/attributes/text/download/all/tag1&&tag2&&!tag3

event_id

Restrict the results to the given event IDs.

allowNonIDS

Allow attributes to be exported that are not marked as "to_ids".

from

Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event.

to

Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event.

last
Events published within the last x amount of time, where x can be defined in days, hours, minutes (for example 5d or 12h or 30m). This filter will use the published timestamp of the event.

The keywords false or null should be used for optional empty parameters in the URL.

For example, to retrieve all attributes for event #5, including non IDS marked attributes too, use the following line:

https://<misp url>/attributes/text/download/all/null/5/true

RESTful searches with JSON result

It is possible to search the database for attributes based on a list of criteria.

To return an event with all of its attributes, relations, shadowAttributes, use the following syntax:

https://<misp url>/attributes/restSearch/json/[value]/[type]/[category]/[org]/[tag]/[quickfilter]/[from]/[to]/[last]/[eventid]/[withAttachments]/[metadata]/[uuid]

If you include "includeEventUuid":1 in the json request, it will give you the event uuid as a result as well.

Be careful if you GET the /attributes/restSearch/json/ without an value, it will return all attributes.

POST /attributes/restSearch

Do not use that function with GET!

Example

```
```

{  "response": []}

RESTful searches with XML result export

It is possible to search the database for attributes based on a list of criteria.

To return an event with all of its attributes, relations, shadowAttributes, use the following syntax:

https://<misp url>/events/restSearch/download/[value]/[type]/[category]/[org]/[tag]/[quickfilter]/[from]/[to]/[last]/[eventid]/[withAttachments]/[metadata]/[uuid]

value
Search for the given value in the attributes' value field.

type
The attribute type, any valid MISP attribute type is accepted.

category
The attribute category, any valid MISP attribute category is accepted.
org
Search by the creator organisation by supplying the organisation identifier.

tags
To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'. You can also chain several tag commands together with the '&&' operator. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead).

For example, to include tag1 and tag2 but exclude tag3 you would use:

```
https://<misp url>/events/restSearch/download/null/null/null/null/tag1&&tag2&&!tag3
```

quickfilter
Enabling this (by passing "1" as the argument) will make the search ignore all of the other arguments, except for the auth key and value. MISP will return an xml / json (depending on the header sent) of all events that have a sub-string match on value in the event info, event orgc, or any of the attribute value1 / value2 fields, or in the attribute comment.

from
Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event.

to
Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event.

last
Events published within the last x amount of time, where x can be defined in days, hours, minutes (for example 5d or 12h or 30m). This filter will use the published timestamp of the event.

eventid
The events that should be included / excluded from the search

withAttachments
Include the attachments/encrypted samples in the export

metadata
Only fetch the event metadata (event data, tags, relations) and skip the attributes

limit
Limit the number of results returned; use together with page.

page
If a limit is set, sets the page to be returned, starting at 1; page 3, limit 100 will return records 201->300). When requesting a page beyond the number of available pages, the returned results list will be empty.

The keywords false or null should be used for optional empty parameters in the URL.

For example, to find any event with the term “red october” mentioned, use the following syntax (the example is shown as a POST request instead of a GET, which is highly recommended):

POST to:

```
https://<misp url>/events/restSearch/download
```

POST message payload (XML):

```
<request>
  <value>red october</value>
  <searchall>1</searchall>
  <eventid>!15</eventid>
</request>
```
POST message payload (JSON):

```json
{"request": {"value": "red october", "searchall": 1, "eventid": ":15"}}
```

To just return a list of attributes, use the following syntax:

- **value**
  - Search for the given value in the attributes' value field.

- **type**
  - The attribute type, any valid MISP attribute type is accepted.

- **category**
  - The attribute category, any valid MISP attribute category is accepted.

- **org**
  - Search by the creator organisation by supplying the organisation identifier.

- **tags**
  - To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'. You can also chain several tag commands together with the '&&' operator. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead).

- **from**
  - Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event.

- **to**
  - Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event.

- **last**
  - Events published within the last x amount of time, where x can be defined in days, hours, minutes (for example 5d or 12h or 30m). This filter will use the published timestamp of the event.

- **eventid**
  - The events that should be included / excluded from the search.

- **uuid**
  - The returned events must include an attribute with the given UUID, or alternatively the event's UUID must match the value(s) passed.

The keywords false or null should be used for optional empty parameters in the URL.

```
https://<misp url>/attributes/restSearch/download/[value]/[type]/[category]/[org]/[tag]/[from]/[to]/[last]/[eventid]/[withattachments]/[uuid]
```

Value, type, category and org are optional. It is possible to search for several terms in each category by joining them with the '&&' operator. It is also possible to negate a term with the '!' operator. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead). For example, in order to search for all attributes created by your organisation that contain 192.168 or 127.0 but not 0.1 and are of the type ip-src, excluding the events that were tagged tag1 use the following syntax:

```
https://<misp url>/attributes/restSearch/download/192.168&&127.0&&!0.1/ip-src/false/CIRCL/!tag1
```
You can also use search for IP addresses using CIDR. Make sure that you use '|' (pipe) instead of '/' (slashes). Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead). See below for an example:


**Export attributes of event with specified type as XML**

If you want to export all attributes of a pre-defined type that belong to an event, use the following syntax:

https://<misp url>/attributes/returnAttributes/download/[id]/[type]/[sigOnly]

sigOnly is an optional flag that will block all attributes from being exported that don't have the IDS flag turned on. It is possible to search for several types with the '&&' operator and to exclude values with the '!' operator. For example, to get all IDS signature attributes of type md5 and sha256, but not filename|md5 and filename|sha256 from event 25, use the following:

https://<misp url>/attributes/returnAttributes/download/25/md5&&sha256&&!filename/true

**Filtering event metadata**

As described in the REST section, it is possible to retrieve a list of events along with their metadata by sending a GET request to the /events API. However, this API in particular is a bit more versatile. You can pass search parameters along to search among the events on various fields and retrieve a list of matching events (along with their metadata). Use the following URL:

https://<misp url>/events/index

POST a JSON object with the desired lookup fields and values to receive a JSON back. An example for a valid lookup:

```
Authorization: <your API key>
Accept: application/json
Content-type: application/json

Body:

{"searchinfo":"Locky", "searchpublished":1, "searchdistribution":0}
```

The list of valid parameters:

- `searchpublished:` Filters on published or unpulished events [0,1] - negatable
- `searchinfo:` Filters on strings found in the event info - negatable
- `searchtag:` Filters on attached tag names - negatable
- `searcheventid:` Filters on specific event IDs - negatable
- `searchthreatlevel:`
Filters on a given event threat level \([1,2,3,4]\) - negatable

**searchdistribution:**
Filters on the distribution level \([0,1,2,3]\) - negatable

**searchanalysis:**
Filters on the given analysis phase of the event \([0,1,2,3]\) - negatable

**searchattribute:**
Filters on a contained attribute value - negatable

**searchorg:**
Filters on the creator organisation - negatable

**searchemail:**
Filters on the creator user's email address (admin only) - negatable

**searchDatefrom:**
Filters on the date, anything newer than the given date in YYYY-MM-DD format is taken - non-negatable

**searchDateuntil:**
Filters on the date, anything older than the given date in YYYY-MM-DD format is taken - non-negatable

**Download attachment or malware sample**

If you know the attribute ID of a malware-sample or an attachment, you can download it with the following syntax:

```
https://<misp url>/attributes/downloadAttachment/download/[Attribute_id]
```

**Download malware sample by hash**

You can also download samples by knowing its MD5 hash. Simply pass the hash along as a JSON/XML object or in the URL (with the URL having overruling the passed objects) to receive a JSON/XML object back with the zipped sample base64 encoded along with some contextual information.

You can also use this API to get all samples from events that contain the passed hash. For this functionality, just pass the "allSamples" flag along. Note that if you are getting all samples from matching events, you can use all supported hash types (md5, sha1, sha256) for the lookup.

You can also get all the samples from an event with a given event ID, by passing along the eventID parameter. Make sure that either an event ID or a hash is passed along, otherwise an error message will be returned. Also, if no hash is set, the allSamples flag will get set automatically.

```
https://attributes/downloadSample/[hash]/[allSamples]/[eventID]
```

POST message payload (XML):

```
<request><hash>7c12772809c1c0c3deda6103b10f6fa8</hash><allSamples>1</allSamples><eventID>13</eventID></request>
```

POST message payload (json):

```
{"request": {"hash": "7c12772809c1c0c3deda6103b10f6fa8", "allSamples": 1, "eventID": 13}}
```

A description of all the parameters in the passed object:
hash
A hash in MD5 format. If allSamples is set, this can be any one of the following: md5, sha1, sha256.

allSamples
If set, it will return all samples from events that have a match for the hash provided above.

eventID
If set, it will only fetch data from the given event ID.

Upload malware samples using the "Upload Sample" API

https://<misp url>/events/upload_sample/[Event_id]

This API will allow you to populate an event that you have modify rights to with malware samples (and all related hashes). Alternatively, if you do not supply an event ID, it will create a new event for you.

The files have to be base64 encoded and POSTed as explained below. All samples will be zipped and password protected (with the password being "infected"). The hashes of the original file will be captured as additional attributes.

For sample upload (for objects in general) there is no check for duplicates.

The event ID is optional. MISP will accept either a JSON or an XML object posted to the above URL.

The general structure of the expected objects is as follows:

```
{"request": {"files": ["filename": filename1, "data": base64encodedfile1], 
{"filename": filename2, "data": base64encodedfile2]},

"optional_parameter1", "optional_parameter2", "optional_parameter3"}]
```

**JSON:**

```
{"request":{"files": ["filename": "test1.txt", "data": "dGVzdA="], 
{"filename": "test2.txt", "data": "dGVzdDII="}], "distribution": 1, "info": "test", "event_id": 15}]
```

**XML:**

```
<request><files><filename>test3.txt</filename><data>dGVzdA==</data></files><files><filename>test4.txt</filename>
<data>dGVzdDII=</data></files><info><test/></info><distribution>1</distribution><event_id>15</event_id></request>
```

The following optional parameters are expected:

**event_id**
The Event's ID is optional. It can be either supplied via the URL or the POSTed object, but the URL has priority if both are provided. Not supplying an event ID will cause MISP to create a single new event for all of the POSTed malware samples. You can define the default settings for the event, otherwise a set of default settings will be used.

**distribution**
The distribution setting used for the attributes and for the newly created event, if relevant. [0-3]

**to_ids**
You can flag all attributes created during the transaction to be marked as "to_ids" or not.

**category**
The category that will be assigned to the uploaded samples. Valid options are: Payload delivery, Artifacts dropped, Payload Installation, External Analysis.
**info**

Used to populate the event info field if no event ID supplied. Alternatively, if not set, MISP will simply generate a message showing that it's a malware sample collection generated on the given day.

**analysis**

The analysis level of the newly created event, if applicable. [0-2] threat_level_id: The threat level ID of the newly created event, if applicable. [0-3]

**comment**

This will populate the comment field of any attribute created using this API.

The threat_level_id is mapped as such:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>high</td>
</tr>
<tr>
<td>1</td>
<td>medium</td>
</tr>
<tr>
<td>2</td>
<td>low</td>
</tr>
<tr>
<td>3</td>
<td>undefined</td>
</tr>
</tbody>
</table>

**Proposals API**

You can interact with the proposals via the API directly since version 2.3.148.

<table>
<thead>
<tr>
<th>HTTP</th>
<th>URL</th>
<th>Explanation</th>
<th>Expected Payload</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/shadow_attributes/view/[proposal_id]</td>
<td>View a proposal</td>
<td>N/A</td>
<td>ShadowAttribute object</td>
</tr>
<tr>
<td>POST</td>
<td>/shadow_attributes/add/[event_id]</td>
<td>Propose a new attribute to an event</td>
<td>ShadowAttribute object</td>
<td>ShadowAttribute object</td>
</tr>
<tr>
<td>POST</td>
<td>/shadow_attributes/edit/[attribute_id]</td>
<td>Propose an edit to an attribute</td>
<td>ShadowAttribute object</td>
<td>ShadowAttribute object</td>
</tr>
<tr>
<td>POST</td>
<td>/shadow_attributes/accept/[proposal_id]</td>
<td>Accept a proposal</td>
<td>N/A</td>
<td>Message</td>
</tr>
<tr>
<td>POST</td>
<td>/shadow_attributes/discard/[proposal_id]</td>
<td>Discard a proposal</td>
<td>N/A</td>
<td>Message</td>
</tr>
</tbody>
</table>

When posting a shadow attribute object, use the following format

**JSON:**

```json
{"request": ["ShadowAttribute": {"value": "5.5.5.5", "to_ids": false, "type": "ip-dst", "category": "Network activity"}"
```

**XML:**

```xml
<request><ShadowAttribute><value>5.5.5.5</value><to_ids>0</to_ids><type>ip-dst</type><category>Network activity</category></ShadowAttribute></request>
```

None of the above fields are mandatory, but at least one of them has to be provided.

**Sharing groups**

MISP allows sharing groups to be retrieved via the API.

https://<misp url>/sharing_groups/index.json
Based on the API key used, the list of visible sharing groups will be returned in a JSON file. The JSON includes the organization parts of a given sharing group along with the associated server.

**Enable, disable and fetching feeds via the API**

The MISP feeds can be enabled via the API.

A feed can be enabled by POSTing on the following URL (feed_id is the id of the feed):

```
/feeds/enable/feed_id
```

A feed can be disabled by POSTing on the following URL (feed_id is the id of the feed):

```
/feeds/disable/feed_id
```

All feeds can cached via the API:

```
/feeds/cacheFeeds/all
```

or you can replace all by the feed format to fetch like misp or freetext. All can be replaced with the id value of the feed to fetch a specific feed.

To fetch a feed or all feeds:

```
/feeds/fetchFromFeed/feed_id
/feeds/fetchFromAllFeeds
```

This API can be also used to download feeds at regular interval via cronjobs or alike.

**Sightings API**

MISP allows Sightings data to be conveyed in several ways.

The most basic way is to POST a blank message to the Sightings API with the attribute ID or attribute UUID. This will create a sightings entry with the creation of the entry as the timestamp for the organisation of the authenticated user.

```
https://<misp url>/sightings/add/[attribute_id]
https://<misp url>/sightings/add/[attribute_uuid]
```

Alternatively, it is possible to POST a JSON object and gain additional granularity. The following fields are recognised by the API:

- **id**
  - The attribute's ID

- **uuid**
  - The attribute's UUID

- **value**
  - Will create a sighting for any attribute with the given value or for composite attributes, for the value matching any element of the attribute value
values

Expects a list, MISP will create sightings for any attribute matching any of the given values or for composite attributes, for any of the values matching any element of the attribute value.

timestamp

Unix timestamp of the sighting, overrides the current time.

Some examples:

To create a sighting for attribute #9001:

```json
{"id": "9001"}
```

To create a sighting for any attribute with the value being teamliquid.net or 173.231.136.216 with the time of sighting being:

```json
{"values": ["teamliquid.net", "173.231.136.216"], "timestamp": 1460558710}
```

It is also possible to POST a STIX indicator with sighting data to the following URL (keep in mind that the content type has to be XML):

https://<misp url>/sightings/add/stix

MISP will use the sightings related observables to gather all values and create sightings for each attribute that matches any of the values. If no related observables are provided in the Sighting object, then MISP will fall back to the Indicator itself and use its observables' values to create the sightings. The time of the sighting is the current time, unless the timestamp attribute is set on the Sightings object, in which case that is taken.

An example STIX sightings document:

```xml
<stix:STIX_Package
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:stix="http://stix.mitre.org/stix-1"
    xmlns:indicator="http://stix.mitre.org/Indicator-2"
    xmlns:stixCommon="http://stix.mitre.org/common-1"
    xmlns:cybox="http://cybox.mitre.org/cybox-2"
    xmlns:AddressObject="http://cybox.mitre.org/objects#AddressObject-2"
    xmlns:DomainNameObj="http://cybox.mitre.org/objects#DomainNameObject-1"
    xmlns:cyboxVocabs="http://cybox.mitre.org/default_vocabularies-2"
    xmlns:stixVocabs="http://stix.mitre.org/default_vocabularies-1"
    xmlns:example="http://example.com/"
    xsi:schemaLocation=
        http://stix.mitre.org/stix-1 ../stix_core.xsd
        http://stix.mitre.org/Indicator-2 ../indicator.xsd
        http://cybox.mitre.org/objects#DomainNameObject-1 http://cybox.mitre.org/XMLSchema/objects/Domain_Name/1.0/
        Domain_Name.xsd
        http://stix.mitre.org/common-1 http://stix.mitre.org/XMLSchema/common/1.1.1/stix_common.xsd
        http://stix.mitre.org/default_vocabulary-2 ../cybox/cybox_default_vocabulary-2.xsd
        http://stix.mitre.org/default_vocabulary-1 ../cybox/cybox_default_vocabulary-1.xsd
        http://cybox.mitre.org/objects#AddressObject-2 ../cybox/objects/Address_Object.xsd"
    id="example:STIXPackage-33fe3b22-0201-47cf-85d0-97c02164528d"
    timestamp="2014-05-08T09:00:00.000000Z"
    version="1.1.1">
    <stix:STIX_Header>
        <stix:Title>Example watchlist that contains IP information.</stix:Title>
        <stix:Package_Intent xsi:type="stixVocabs:PackageIntentVocab-1.0">Indicators - Watchlist</stix:Package_Intent>
    </stix:STIX_Header>
    <stix:Indicators>
        <stix:Indicator xsi:type="indicator:IndicatorType" id="example:Indicator-2e28c5b2-56fa-46cd-9662-8f199c699d2c9" timestamp="2014-05-08T09:09:00.000000Z">
            <cybox:AddressObject>
                <cybox:AddressObjectProperties>
                    <cyboxv:Property id="example:AddressObject-voc-2629">
                        <cyboxv:PropertyValues>
                            <cybox:AddressValue>173.231.136.216</cybox:AddressValue>
                            <cybox:AddressValue>teamliquid.net</cybox:AddressValue>
                        </cyboxv:PropertyValues>
                    </cyboxv:Property>
                </cybox:AddressObjectProperties>
            </cybox:AddressObject>
            <stixCommon:Confidence id="example:Confidence-voc-6369">2</stixCommon:Confidence>
            <stixCommon:Last_Updated id="example:Last_Updated-voc-7520">2014-05-08T09:09:00Z</stixCommon:Last_Updated>
        </stix:Indicator>
    </stix:Indicators>
</stix:STIX_Package>
```
POSTing this as the message’s body to MISP will sight any attributes visible to the user with he value “malicious2.example.com”. For composite types, a match on a component will also trigger a sighting (so for example for attributes of type domain|ip a domain match would be sufficient).

If no Related observables are set in the Sighting itself, MISP will fall back to the observable directly contained in the indicator. So in the following example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<stix:STIX_Package
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmli="http://stix.mitre.org/fixtures/stix-package.xsd"
xmli="http://www.w3.org/2001/XMLSchema"
xmlns:stix="http://stix.mitre.org/stix-1"
xmlns:indicator="http://stix.mitre.org/Indicator-2"
xmlns:stixCommon="http://stix.mitre.org/common-1"
xmlns:cybox="http://cybox.mitre.org/cybox-2"
xmlns:AddressObject="http://cybox.mitre.org/objects#AddressObject-2"
xmlns:DomainNameObject="http://cybox.mitre.org/objects#DomainNameObject-1"
xmlns:cyboxVocabs="http://cybox.mitre.org/default_vocabularies-2"
xmlns:stixVocabs="http://stix.mitre.org/default_vocabularies-1"
xmlns:example="http://example.com/"
xmlns:xsi:schemaLocation="http://stix.mitre.org/stix-1 ...
.../stix_core.xsd"
http://stix.mitre.org/Indicator-2 ...
.../indicator.xsd"
http://cybox.mitre.org/objects#DomainNameObject-1 http://cybox.mitre.org/XMLSchema/objects/Doma"

Domain_Name_Object.xsd
http://stix.mitre.org/common-1 http://stix.mitre.org/XMLSchema/common/1.1.1/stix_common.xsd
http://cybox.mitre.org/default_vocabularies-2 ...
.../cybox_default_vocabularies.xsd
http://stix.mitre.org/default_vocabularies-1 ...
.../stix_default_vocabularies.xsd
http://cybox.mitre.org/objects#AddressObject-2 ...
.../cybox/objects/Address_Object.xsd"

id="example:STIXPackage-33f6bd22-0261-47cf-856b-97c26264528d"

timestamp="2014-05-08T09:00:00.000000Z"

version="1.1.1" />

<stix:STIX_Head>
<stix:Title>Example watchlist that contains IP information.</stix:Title>

```

Automation and MISP API
MISP would create sightings for attributes matching any of the following: malicious1.example.com, malicious2.example.com, malicious3.example.com

**Warninglists API**

**GET warninglists/index**

**Description**

Return the index of warninglists enabled on the MISP instance

**Parameters**

- id

**Output**

```json
...
["Warninglists":[{"id":"17","name":"List of known Office 365 URLs and IP address ranges","type":"string","description":"Office 365 URLs and IP address ranges","enabled":true,"warninglist_entry_count":1516,"valid_attributes":[]}],
{"id":"19","name":"List of hashes for EICAR test virus","type":"string","description":"Event contains one or more entries based on hashes for EICAR test virus","enabled":true,"warninglist_entry_count":665,"valid_attributes":[]},
{"id":"14","name":"Top 1000 website from Alexa","type":"string","description":"Event contains one or more entries from the top 1000 of the most used website (Alexa).","enabled":true,"warninglist_entry_count":2014-05-08T09:00:00.000000Z","valid_attributes":[]}
```
create an aggregates list. To access the attribute type or attribute category data distribution on your instance, MISP offers an API that will create an aggregates list. To access the API, simple sent a GET request to:

https://<misp url>/attributes/attributeStatistics/[context]/[percentage]
Where the following parameters can be set:

**Context**
Set whether you are interested in the type or category statistics of your instance. This parameter can be either set to "type" or "category", with type being the default setting if the parameter is not set.

**Percentage**
An optional field, if set, it will return the results in percentages instead of the count.

The results are always returned as JSON.

Sample output of the types in percentages from CIRCL’s MISP instance:

```json
{
  "AS": "0.015%",
  "attachment": "0.177%",
  "btc": "0.005%",
  "campaign-name": "0.005%",
  "comment": "1.47%",
  "domain": "15.992%",
  "domain|ip": "0.005%",
  "email-attachment": "0.207%",
  "email-dist": "0.123%",
  "email-src": "0.192%",
  "email-subject": "0.140%",
  "filename": "3.698%",
  "filename|md5": "0.349%",
  "filename|sha1": "0.045%",
  "filename|sha256": "0.052%",
  "hostname": "17.558%",
  "http-method": "0.045%",
  "ip-dist": "7.887%",
  "ip-src": "2.787%",
  "link": "5.748%",
  "malware-sample": "0.702%",
  "malware-type": "0.085%",
  "md5": "21.064%",
  "mutex": "0.278%",
  "named\tpipe": "0.03%",
  "other": "1.499%",
  "pattern-in-file": "0.192%",
  "pattern-in-memory": "0.303%",
  "pattern-in-traffic": "0.051%",
  "regkey": "0.120%",
  "regkey\value": "0.187%",
  "sha1": "8.921%",
  "sha256": "5.597%",
  "snort": "0.045%",
  "target-machine": "0.248%",
  "target-org": "0.01%",
  "target-user": "0.106%",
  "text": "0.934%",
  "threat-actor": "0.005%",
  "url": "2.258%",
  "user-agent": "0.081%",
  "vulnerability": "0.182%",
  "whois-registrant-email": "0.01%",
  "x509-fingerprint-sha1": "0.01%",
  "yara": "0.086%"
}
```
Additional statistics are available as JSON which are the statistics also usable via the user interface. A ".json" can be appended to the following URLs:

- https://<misp url>/users/statistics/tags.json
- https://<misp url>/users/statistics.json
- https://<misp url>/users/statistics/attributehistogram.json
- https://<misp url>/users/statistics/orgs.json

An example output of https://users/statistics.json:

```
{
  "stats": {
    "event_count": 5233,
    "event_count_month": 21,
    "attribute_count": 645498,
    "attribute_count_month": 723,
    "correlation_count": 207152,
    "proposal_count": 48944,
    "user_count": 1073,
    "org_count": 587,
    "thread_count": 191,
    "thread_count_month": 0,
    "post_count": 337,
    "post_count_month": 0
  }
}
```

### MISP modules

#### Description

It is possible call misp-modules directly from API. If the module needs credentials, API will get the information directly from MISP configuration.

#### GET /modules/

Retrieve a list of all modules enabled.

**Example**

```
```

**Output**

```
[
  {
    "name": "passivetotal",
    "type": "expansion",
    "mispattributes": {
      "input": [
        "hostname",
        "domain",
        "ip-src",
        "ip-dst"
      ]
  }
]```
POST /modules/queryEnrichment

Call any enabled module.

Example

Content of dns.json

{  
  "output": [  
    "ip-src",  
    "ip-dst",  
    "hostname",  
    "domain"
  ],  
  "meta": {  
    "description": "Simple DNS expansion service to resolve IP address from MISP attributes",  
    "author": "Alexandre Dulaunoy",  
    "version": "0.1"
  }
}
"hostname": "www.foo.be",
"module": "dns"
}

Query using MISP API


The output will be following JSON:

{
   "results": [
      {
         "types": [
            "ip-src",
            "ip-dst"
         ],
         "values": [
            "188.65.217.78"
         ]
      }
   ]
}
PyMISP - Python Library to access MISP

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.

Note that you need to have Auth Key access in your MISP instance to use PyMISP

Capabilities

- Add, get, update, publish, delete events
- Add or remove tags
- Add file attributes: hashes, registry key, patterns, pipe, mutex
- Add network attributes: IP dest/src, hostname, domain, url, UA, ...
- Add Email attributes: source, destination, subject, attachment, ...
- Upload/download samples
- Update sightings
- Proposals: add, edit, accept, discard
- Full text search and search by attributes
- Get STIX event
- Export statistics And even more, just look at the api.py file

Installation

You can install PyMISP by either using pip or by getting the last version from the GitHub repository

Install from pip

```
pip install pymisp
```

Install the latest version from the repository

```
git clone https://github.com/MISP/PyMISP.git && cd PyMISP
git python setup.py install
```

Note that you will also need to install requests if you don't have it already.

Getting started

You now need to get your automation key. You can find it on the automation page:

```
https://<misp url>/events/automation
```

or on your profile

```
https://<misp url>/users/view/me
```

If you did not install using the repository, you can still fetch it to get examples to work on:
In order to use these, you need to create a file named keys.py in the examples folder and edit it to put the url of your MISP instance and your automation key.

```bash
cd examples
cp keys.py.sample keys.py
vim keys.py
```

Once you are done with it, you are ready to start.

This is how `keys.py` looks:

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-

misp_url = 'https://'
misp_key = 'Your MISP auth key'  # The MISP auth key can be found on the MISP web interface under the automation section
misp_verifycert = True
```

## Using PyMISP

To have a better understanding of how to use PyMISP, we will have a look at one of the existing examples:

- `add_named_attribute.py` This script allows us to add an attribute to an existing event while knowing only its type (the category is determined by default).

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from pymisp import PyMISP
from keys import misp_url, misp_key
import argparse
```

First of all, it is obvious that we need to import PyMISP. Then we also need to know both the instance with which we will work and the API key to use: Both should be stored in the keys.py file. Finally, we import argparse library so the script can handle arguments.

```python
# For python2 & 3 compat, a bit dirty, but it seems to be the least bad one
try:
    input = raw_input
except NameError:
    pass
```

Just a few lines to be sure that python 2 and 3 are supported

```python
def init(url, key):
    return PyMISP(url, key, True, 'json', debug=True)
```

This function will create a PyMISP object that will be used later to interact with the MISP instance. As seen in the `api.py`, a PyMISP object need to know both the URL of the MISP instance and the API key to use. It can also take additional and not mandatory data, such as the use or not of SSL or the name of the export format.

```python
if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Create an event on MISP.')
    parser.add_argument('-e', '--event', type=int, help='The id of the event to update.')
```
Then the function starts by preparing the awaited arguments:

- event: The event that will get a new attribute
- type: The type of the attribute that will be added. See here for more information
- value: The value of the new attribute

```python
misp = init(misp_url, misp_key)
```

Thanks to the previously created function, we create a PyMISP object.

```python
event = misp.get_event(args.event)
event = misp.add_named_attribute(event, args.type, args.value)
```

In order to add the new argument, we first need to fetch the event in the MISP database using the `get_event` function which only need the event_id. Then only once we have it, we can call the function `add_named_attribute` that will add the argument.

```python
print(event)
```

Finally the new event is printed, so we can check that the attribute was correctly added, and that a category was attached to it automatically.

### Existing examples

As the name implies you will find several example scripts in the examples folder. For each you can get help if you do `scriptname.py -h`

Let us have a look at some of these examples:

**add_named_attribute.py**

Allow to add an argument to an existing event by giving only the type of the attribute. The category will be set with a default value.

Arguments:

- **event**: The id of the event to update.
- **type**: The type of the added attribute.
- **value**: The value of the attribute.

**add_user.py**

Allow to add a user by giving the mandatory fields as entries.

Arguments:

- **email**: Email linked to the account.
- **org_id**: Organisation linked to the user.
- **role_id**: Role linked to the user.

**add_user_json.py**

Add the user described in the given json. If no file is provided, returns a json listing all the fields used to describe a user.
Arguments:

- **json_file**: The name of the json file describing the user you want to create.

**create_events.py**

Allow a user to create a new event on the MISP instance.

Arguments:

- **distrib**: The distribution setting used for the attributes and for the newly created event, if relevant. [0-3].
- **info**: Used to populate the event info field if no event ID supplied.
- **analysis**: The analysis level of the newly created event, if applicable. [0-2]
- **threat**: The threat level ID of the newly created event, if applicable. [1-4]

**del.py**

Delete an event or an attribute from a MISP instance. The event has the priority: if both are set, only the event will be deleted.

Arguments:

- **event**: Event ID to delete.
- **attribute**: Attribute ID to delete.

**delete_user.py**

Delete the user with the given id. Keep in mind that disabling users (by setting the disabled flag via an edit) is always preferred to keep user associations to events intact.

Arguments:

- **user_id**: The id of the user you want to delete.

**edit_user.py**

Edit the email of the user designed by the user_id.

Arguments:

- **user_id**: The name of the json file describing the user you want to modify.
- **email**: Email linked to the account.

**edit_user_json.py**

Edit the user designed by the user_id. If no file is provided, returns a json listing all the fields used to describe a user.

Arguments:

- **user_id**: The name of the json file describing the user you want to modify.
- **json_file**: The name of the json file describing your modifications.

**get.py**

Get an event from a MISP instance in json format.

Arguments:

- **event**: Event ID to get.
- **output**: Output file

**last.py**

Download latest events from a MISP instance. A output file can be created to store these events.

**Arguments:**
- **last**: can be defined in days, hours, minutes (for example 5d or 12h or 30m).
- **output**: Output file

**searchall.py**

Get all the events matching a value.

**Arguments:**
- **search**: String to search.
- **quiet**: Only display URLs to MISP
- **output**: Output file

**sharing_groups.py**

Get a list of the sharing groups from the MISP instance. No argument.

**sighting.py**

Add sighting.

**Arguments:**
- **json_file**: The name of the json file describing the attribute you want to add sighting to.

**stats.py**

Output attributes statistics from a MISP instance. No argument.

**suricata.py**

Download Suricata events.

**Arguments:**
- **all**: Download all suricata rules available.
- **event**: Download suricata rules from one event.

**tags.py**

Get tags from MISP instance. No argument.

**tagstatistics.py**

Get statistics from tags.

**Arguments:**
- **percentage**: An optional field, if set, it will return the results in percentages, otherwise it returns exact count.
- **namesort**: An optional field, if set, values are sort by the namespace, otherwise the sorting will happen on the value.

**up.py**

Update an existing event regarding the data inside a given json file.

Arguments:
- **event**: Event ID to modify.
- **input**: Input file

**upload.py**

Send malware sample to MISP.

Arguments:
- **upload**: File or directory of files to upload.
- **event**: Not supplying an event ID will cause MISP to create a single new event for all of the POSTed malware samples.
- **distrib**: The distribution setting used for the attributes and for the newly created event, if relevant. [0-3].
- **ids**: You can flag all attributes created during the transaction to be marked as "to_ids" or not.
- **categ**: The category that will be assigned to the uploaded samples. Valid options are: Payload delivery, Artefacts dropped, Payload Installation, External Analysis.
- **info**: Used to populate the event info field if no event ID supplied.
- **analysis**: The analysis level of the newly created event, if applicable. [0-2]
- **threat**: The threat level ID of the newly created event, if applicable. [1-4]
- **comment**: Comment for the uploaded file(s).

**users_list.py**

Get a list of the sharing groups from the MISP instance. No argument.

**Going further**

**feed-generator**

It is used to generate the CIRCL OSINT feed. This script export the events as json, based on tags, organisation, events, ... It automatically update the dumps and the metadata file.

Here is an example of a config file:

```python
url = ''
key = ''
ssl = True
outputdir = 'output'
# filters = {'tag': 'tlp:white|feed-export|privint', 'org': 'CIRCL'}
filters = {}
valid_attribute_distribution_levels = ['0', '1', '2', '3', '4', '5']
```

**Consuming feed**

As the feed is a simple set of MISP json files, the files can be easily imported directly into any MISP instance. The script below processes the manifest file of an OSINT feed and reimport them in a MISP directly.
import requests

url = 'https://www.circl.lu/doc/misp/feed-osint/
osintcircl = requests.get('{}manifest.json'.format(url))
misp = PyMISP('http://misp.test/', 'key', False, 'json')
for uri in osintcircl.json():
    req = requests.get('{}{}.json'.format(url, uri))
misp.add_event(req.json())

ioc-2-misp

Allow to import OpenIOC files into MISP easily. It is also possible to set specific tags on these events.

Situational Awareness

- attribute_treemap.py generate a tree-map showing the distribution of the attributes on the MISP instance.
- tags_*: these functions help having statistics and graphs about the tag repartition.

Simple example on fetching the last events

# Usage for pipe masters: ./last.py -l 5h | jq .

def init(url, key):
    return PyMISP(url, key, misp_verifycert, 'json')

def download_last(m, last, out=None):
    result = m.download_last(last)
    if out is None:
        if 'response' in result:
            print(json.dumps(result['response']))
        else:
            print('No results for that time period')
            exit(0)
    else:
        with open(out, 'w') as f:
            f.write(json.dumps(result['response']))

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Download latest events from a MISP instance.')
    parser.add_argument('-l', '--last', required=True, help="can be defined in days, hours, minutes (for example 5d or 12h or 30m).")
    parser.add_argument('-o', '--output', help="Output file")
    args = parser.parse_args()
    if args.output is not None and os.path.exists(args.output):
print('Output file already exists, abord.\n')
exit(0)

misp = init(misp_url, misp_key)

download_last(misp, args.last, args.output)
Create an event based on a report

[warning] A specific permission is required to create an event.

For this example, we will use a report found on Bleeping Computer, so considered as OSINT.

---

**Researcher finds the Karma Ransomware being distributed via Pay-per-Install Network**

By Lawrence Abrams

November 14, 2016 07:01 PM 2

A security researcher named slipstream/RoL has discovered the Karma Ransomware, which pretends to be a Windows optimization program called Windows-TuneUp. What is worse is that this sample was discovered as software that would potentially be distributed by a pay-per-install software monetization company when people install free software downloaded from the Internet.
Adding an event

First of all, we need to create a new event. To do so, we click the "Add Event" option when on the Events list view.

**List Events**
- Add Event
- Import From MISP Export
- List Attributes
- Search Attributes
- View Proposals
- Events with proposals
- Export
- Automation
Then we get the add event form.

![Add Event Form](image)

- **Date**: 2016-11-16
- **Distribution**: All communities
- **Threat Level**: High
- **Analysis**: Initial
- **Event Info**: Quick Event Description or Tracking Info
- **GFI sandbox**: Browse... No file selected.
- **Add**
Let's fill it with the data we already have:

- **Date**: Here we will put the date of the report, so 2016-11-14.
- **Distribution**: Depending on the event, we might want it to be more or less spread across the MISP instances. For this one, since it is a public report, there is no reason to limit the diffusion so "All communities".
- **Threat Level**: Self explanatory. Since the ransomware in the report is not using a huge exploit, we can use low, or undefined as we don't really know. We'll go for the latter since it can be edited.
- **Analysis**: Give the current stage of the analysis. Since the report is published, we can assume that the analysis is completed.
- **Event Info**: The event's info is in fact the name or title of the event, so it seems legit to put the title of the report here as well. Since it is public information, we also prefix it with "OSINT".
- **GFI sandbox**: Since we don't have any sample or anything here, we leave this alone.

![Add Event Form](image)

<table>
<thead>
<tr>
<th>Date</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016-11-14</td>
<td>All communities</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Threat Level</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undefined</td>
<td>Completed</td>
</tr>
</tbody>
</table>

**Event Info**: OSINT - Researcher finds the Karma Ransomware being distributed via...

**GFI sandbox**: Browse... No file selected.
Then just press the blue "Add" button and here we have a brand new event. Empty.
(Displayed information can change depending on your role on the MISP instance)

**Adding Attributes**

Now it is time to populate this event. But before even adding IoC, we are going to add global information about the report itself: the link of the report and a short explanation or introduction. To do so, we need to click on the "Add Attribute" option in the side menu. This will show us this view:
First we are going to add the link of the report. Since it has been written by an other researcher, it will be considered as an "External analysis", we choose this category.

Concerning the type, regarding the kind of data we are adding it is obvious that we will choose the "link" type.

The distribution field can be a little tricky. We can either choose one of the option that was already available at event level or "Inherit event". If we choose the latter, the attribute will be shared the same way as the event it is included in (here to "All communities"). On the other hand, if we choose manually a distribution for the attribute, the most restrictive between event distribution and attribute distribution will be applied. That is to say: if both event and attribute distributions are the same, there will be no change (similar to "Inherit event"). However, if for instance the event distribution is "all communities" while the attribute is limited to "This community only", the event will indeed be distributed to all communities but without this particular attribute which will be limited to this community only. The same works the other way around, if the attribute can be distributed to "all communities" while the related event is limited to this community, the attribute being dependant of the event, it will be shared to this community only, basing its distribution on the event (most restrictive) one.

The value is simply the data we want to add, here it is the link of the report.

The contextual comment is a field that will not be used for correlation and is mainly there to add some complementary information on the attribute. Can be a port for an IP, or an indication of any type. Here there is no particular information to add, except maybe tell that it is the source of the report, so let us put this information.

"for Intrusion Detection System" is used to set the IDS flag or not. If set, the attribute will be used as an IDS signature when exporting the NIDS data. In this case, we have no reason to check it.

The Batch Import is a useful option when we need to add several IoC of the same category/type which allow you to add them at once by separated by a line break between each line in the value field. However it is of no use here.

Add Attribute

<table>
<thead>
<tr>
<th>Category</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>External analysis</td>
<td>link</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inherit event</td>
</tr>
</tbody>
</table>

Value


Contextual Comment

Source Report

for Intrusion Detection System Batch Import

Submit
All fields are properly filled? Then let's press the "submit" button, and Ta-dah!
Now we can do a similar procedure to add an introduction to the report (that is to say the first paragraph of the report). We will simply change the type for text. But this time, we will access the add attribute form by clicking on the small + symbol next to the attribute table.

Create an Event Based on a Report
The same form as before will appear in a popup.
Again, we fill it with the required data.

A security researcher named slipstream/BoL has discovered the Karma Ransomware, which pretends to be a Windows optimization program called Windows TuneUp. What is worse is that this sample was discovered as software that would potentially be distributed by a pay-per-install software monetization company when people install free software downloaded from the Internet.
Then we submit it by clicking on the blue button *Et voilà!*
Okay, now it is time to add some Indicators of Compromise. In this report, they are mainly listed at the end.

### Files associated with the Karma Ransomware

Windows-TuneUp.exe

### Registry entries associated with the Karma Ransomware

- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer "auth"
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run "Saffron"= "%Desktop%\# DECRYPT MY FILES #.html"
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run "Saffron"= "%Desktop%\# DECRYPT MY FILES #.txt"

### IOCs:

SHA256: 6545ee2b8b11094ad257a7ff25b1ebc63cfc66e742fa76fd44bbd05b74fe0
SHA256: c5f6da20f8e1f135aa68890ca729be60be2cb93888e3f94c0cd9b13f5bc4092

### Network Communication:

- karnotx2gge6ccmudp.onion
- windows-tuneup.com/web293/xUser.php
Let's try to define which category/type those IoC belong to.

First, Windows-TuneUp.exe is without a doubt a filename, and the associated category may be Payload delivery.

Second the registry entries (type regkey) seems to be from Artifacts dropped category

Then the hashes that are already said to be SHA 256, and a quick test on VirusTotal also reveals that they correspond to the filename seen earlier. so we can add both as an association filename\SHA256. Once again, the category will be Payload delivery.

And finally the network communication. No doubt here for the category: Network activity, and the type might be url but for the example, we will let MISP decide for us.

So we begin with the filename. No real change from before for this one, except that we will set the IDS flag to true.
**Freetext Import Tool**

Then we can add the hashes in a similar way. We will add them both alone and combined with the filename. In order to do it quickly, we are going to use the freetext import tool, hidden there.
It will open a popup with a text area field where we will paste our **IoC**, one per line. As said previously, we add both the hashes alone and with the filename.

![Freetext Import Tool](image)
Then when we press the submit button, we are redirected on this page to control the sent data.
Here, MISP detected by itself what should be the category and type associated to our IoC and surprise! It matches our suppositions. Plus, it also put the IDS flag, so it is perfect. But before submitting, please double check to be sure all the values are correct and no information was lost (That can happen when the data are not formatted as expected by MISP).

If the results of MISP were not what we expected, we can still modify it, however MISP will only suggest suitable category/type regarding the format of your data. We can change for each attribute individually or all at the same time using the option on the bottom right of the form. The same principle also applies for the comments, individually or for all.
(Yes I have two cursors, MISP is magic!)

We only have the network **indicators** left, and as said before, we will let MISP determined for us which type is the best for the data we have.
Freetext 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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Similar Attributes</th>
<th>Category</th>
<th>Type</th>
<th>IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>karma2xgg6ccmudp.onion</td>
<td></td>
<td>Payload delivery</td>
<td>filename</td>
<td>✔</td>
</tr>
<tr>
<td>windows-tuneup.com/web293/xUser.php</td>
<td></td>
<td>Network activity</td>
<td>uri</td>
<td>✔</td>
</tr>
</tbody>
</table>

Submit Update all comment fields
Oh well, that was unexpected. In fact, it is not that surprising regarding the format of the tor address that look more like a filename than like a url but it is still a problem, since we can't change the type nor the category to a more consistant one. This is indeed one of the limitation of freetext import. To solve this issue, we will use a simple trick: we will add a slash at the end of the tor address so it won't be confused for a filename.
Create an Event Based on a Report

## Freetext 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 included.

<table>
<thead>
<tr>
<th>Value</th>
<th>Similar Attributes</th>
<th>Category</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>karma2xgg6ccmupd.onion/</td>
<td></td>
<td>Network activity</td>
<td>url</td>
</tr>
<tr>
<td>windows-tuneup.com/web293/xUser.php</td>
<td></td>
<td>Network activity</td>
<td>url</td>
</tr>
</tbody>
</table>
Thanks to the added character, the first string is recognised as an url which is more consistent with the reality. The second also seems okay, so we can now submit both.

**Batch Import**

The Freetext Import works properly only with a string of data without any spaces in one line. But if you have lines of text with spaces between values, like e.g.

**Associated Files:**

```plaintext
%Appdata%\Microsoft\Windows\Start Menu\Programs\Startup\Info.hta
%Appdata%\Microsoft\Windows\Start Menu\Programs\Startup\cmb_ransomware.exe
%Appdata%\Info.hta
%UserProfile%\Desktop\FILES ENCRYPTED.txt
C:\Users\Public\Desktop\FILES ENCRYPTED.txt
```
you can still import them at once using the "Add Attribute" option. Click on Add Attribute, copy the data and paste it into the Value box. Choose the right category and type. Now check both checkboxes for Intrusion Detection System and Batch Import. The option Batch Import will import your data line for line just like the Freetext Import option without losing any information. Like this:

![Add Attribute form]

**Did you consider adding an object instead of a composite attribute?**

**Category**

- Artifacts dropped
- regkey\value

**Distribution**

- Inherit event

**Value**

- %Appdata%\Microsoft\Windows\Start Menu\Programs\Startup\Info.txt
- %Appdata%\Microsoft\Windows\Start Menu\Programs\Startup\cmd_ransomware.exe
- %Appdata%\Info.txt
- C:\Users\Profile\Desktop\FILES ENCRYPTED.txt

**Contextual Comment**

**Notices:**

**[gdpr]**: This attribute is likely to contain personal data and the data subject could be potentially directly identifiable. Please verify where applicable, please ensure that you have taken the necessary steps to ensure transparency towards the data subjects in
And that is all we can get for the main informations and IoC in this report. If we search more carefully, there might still be some information left in it, like the filename of the ransomnote for instance, but we will stop here for this example.

**Modify the event**

If you want to modify your event from the home page, you can either double click on the event or click the edit symbol located in the column *Actions* on the right side. You will be redirected to the editing mode of the selected event.
Taxonomies

In MISP 2.4.X, a flexible mechanism has been introduced to support various taxonomy of classification.

You can access the taxonomy by going into 'Event Actions' and select 'List Taxonomies'. For fresh install, make sure to click 'Update Taxonomies' to view available taxonomies.

A complete list of the available taxonomies PDF are available on the MISP project website.
The following taxonomies can be used in MISP (as local or distributed tags) or in other tools willing to share common taxonomies among security information sharing tools.

**MISP taxonomies - Flexible Classification for Information Sharing**

MISP taxonomies is a solution to use existing taxonomies (or create your own) to classify your cybersecurity events, indicators and threats. This technique is integrated as a default mechanism for tagging in MISP (Malware Information Sharing Platform & Threat Sharing) and to support a distributed classification where organizations can share common taxonomies in a local or distributed fashion.

Classifications are distributed as simple JSON files to use with MISP but can be easily integrated into any other information sharing software. You can also propose new taxonomies to the community.

### Examples of machine tags and human readable tags:

- **admiralty-scale:**
  - `source-reliability="c"` ➔ `Source Reliability = Fairly reliable`
  - `information-credibility="3"` ➔ `Information Credibility = Possibly true`

- **nato:**
  - `classification="NU"` ➔ `NATO UNCLASSIFIED`
  - `nato:classification="NU"` ➔ `NATO UNCLASSIFIED`

- **tlp:**
  - `amber` ➔ `Traffic Light Protocol (TLP): AMBER` Information exclusively given to an organization; sharing limited within the organization to be effectively acted upon.

[namespace][predicate][value]

[https://github.com/MISP/misp-taxonomies/](https://github.com/MISP/misp-taxonomies/)
1. **Admiralty Scale**: The Admiralty Scale (also called the NATO System) is used to rank the reliability of a source and the credibility of an information.

2. **adversary** An overview and description of the adversary infrastructure.

3. **CIRCL Taxonomy - Schemes of Classification in Incident Response and Detection** CIRCL Taxonomy is a simple scheme for incident classification and area topic where the incident took place.

4. **Cyber Kill Chain** from Lockheed Martin as described in *Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains*.

5. **DE German (DE) Government classification markings (VS)** Taxonomy for the handling of protectively marked information in MISP with German (DE) Government classification markings (VS).


7. **Diamond Model for Intrusion Analysis**, a phase-based model developed by Lockheed Martin, aims to help categorise and identify the stage of an attack.

8. **Domain Name Abuse** - taxonomy to tag domain names used for cybercrime. Use europol-incident to tag abuse-activity

9. **eCSIRT** eCSIRT incident classification Appendix C of the eCSIRT EU project including IntelMQ updates.

10. **ENISA ENISA Threat Taxonomy** - A tool for structuring threat information as published

11. **Estimative Language** Estimative language - including likelihood or probability of event based on the Intelligence Community Directive 203 (ICD 203) (6.2.(a)).

12. **[EU Marketop and Publicadmin]** EU critical sectors Market operators and public administrations that must comply to some notifications requirements under EU NIS directive.

13. **EUCI** EU classified information (EUCI) means any information or material designated by a EU security classification, the unauthorised disclosure of which could cause varying degrees of prejudice to the interests of the European Union or of one or more of the Member States as described.

14. **Europol Incident** EUROPOL class of incident taxonomy

15. **Europol Events** - EUROPOL type of events taxonomy

16. **FIRST CSIRT Case** FIRST CSIRT Case Classification.

17. **FIRST Information Exchange Policy (IEP)** framework

18. **French gov information classification system**

19. **Information Security Indicators** Information security indicators have been standardized by the ETSI Industrial Specification Group (ISG) ISI. These indicators provide the basis to switch from a qualitative to a quantitative culture in IT Security Scope of measurements: External and internal threats (attempt and success), user's deviant behaviours, nonconformities and/or vulnerabilities (software, configuration, behavioural, general security framework).

20. **Information Security Marking Metadata (ISM) V13** as described by DNI.gov.

21. **Malware** classification based on different categories. Based on a SANS whitepaper about malware.

22. **Malware Type and Platform classification** based on Microsoft's implementation of the Computer Antivirus Research Organization (CARO) Naming Scheme and Malware Terminology. Based on Microsoft Malware naming conventions, Microsoft Glossary, Microsoft Objective Criteria, and CARO’s definitions. Malware families are extracted from Microsoft SIRs since 2008 based on Microsoft Malware, virus, and threat encyclopedia. Note that SIRs do NOT include all Microsoft malware families.

23. **MISP taxonomy** to infer with MISP behavior or operation.
24. **ms-caro-malware** Malware Type and Platform classification based on Microsoft's implementation of the Computer Antivirus Research Organization (CARO) Naming Scheme and Malware Terminology.

25. **NATO Classification Marking** Marking of Classified and Unclassified materials as described by the North Atlantic Treaty Organization, NATO.


27. **OSINT Open Source Intelligence - Classification**

28. **The Permissible Actions Protocol - or short: PAP** PAP was designed to indicate how the received information can be used. It's a protocol/taxonomy similar to TLP informing the recipients of information what they can do with the received information.

29. **Status of events used in Request Tracker.**

30. **Classification based on malware stealth techniques.** Described in Introducing Stealth Malware Taxonomy

31. **TLP - Traffic Light Protocol** The Traffic Light Protocol - or short: TLP - was designed with the objective to create a favorable classification scheme for sharing sensitive information while keeping the control over its distribution at the same time.

32. **Vocabulary for Event Recording and Incident Sharing** VERIS

A taxonomy contains a series of tags that can be used as normal tags in your MISP instance.

**Tagging** is a simple way to attach a classification to an event. In the early version of MISP, tagging was local to an instance. Classification must be globally used to be efficient. After evaluating different solutions of classification, we build a new scheme using the concept of machine tags.

Taxonomy is a classification of informations. Taxonomies are implemented in a simple JSON format. Anyone can create their own taxonomy or reuse an existing one.

Taxonomies are in an independent git repository.

These can be **freely reused** and **integrated** in other threat intel tools.

The advantage is that you can set a specific tag as being exportable. This means that you can export your classification with other MISP instance and share the same taxonomies. Tagging is a simple way to attach a classification to an event.

**Classification must be globally used to be efficient.**

If you want to enable a specific taxonomy, you can click on the cross to enable it.
Then you can even cherry-pick the tags you want to use on the system. If you want to use the whole taxonomy, select all and then click on the cross in the top left.

**Contributing to Taxonomy**

It is quite easy. Create a JSON file describing your taxonomy as triple tags.

```json
{
   "namespace": "admiralty-scale",
   "description": "The Admiralty Scale (also called the NATO System) is used to rank the reliability of a source and the credibility of an information.",
   "version": 1,
   "predicates": [
      "source-reliability",
      "expanded": "Source Reliability"
   ],
   "information-credibility",
   "expanded": "Information Credibility"
}
```
{ "values": [
  { "predicate": "source-reliability",
    "entry": [ 
      { "value": "a",
        "expanded": "Completely reliable"
      }
    ]
  }
]
(e.g. check an existing one like Admiralty Scale), create a directory matching your name space, put your machinetag file in the directory and pull your request. Publishing your taxonomy is as easy as a simple git pull request on misp-taxonomies (https://github.com/MISP/misp-taxonomies). That's it. Everyone can benefit from your taxonomy and can be automatically enabled in information sharing tools like MISP.

Reserve Taxonomy

The following taxonomy namespaces are reserved and used internally to MISP.

- galaxy mapping taxonomy with cluster:element:"value".

Adding Taxonomy in MISP

How are taxonomies integrated in MISP?

MISP administrators have only to import (or even cherry pick) the namespace or predicates they want to use as tags.

Tags can be exported to other instances.

Tags are also accessible via the MISP REST API.

For more information, "Information Sharing and Taxonomies Practical Classification of Threat Indicators using MISP" presentation given to the last MISP training in Luxembourg.

Adding a private taxonomy

```
$ cd /var/www/MISP/app/files/taxonomies/
$ mkdir privatetaxonomy
$ cd privatetaxonomy
$ vi machinetag.json
```

Create a JSON file describing your taxonomy as triple tags.

For example:

```
mkdir sample
cd sample
vim machinetag.json
```

Sample JSON with triple tags. You can use the JSON validator to be sure that there is no syntax error.

```json
{
   "namespace": "sample",
   "description": "Some descriptive words",
   "version": 1,
   "predicates": [ {
                     "value": "my-predicate",
                     "expanded": "my-predicate"
                  }
   ]
}
```

---

260
Go to MISP Web GUI taxonomies/index and update the taxonomies once you are happy with your file. The newly created taxonomy should be visible. Now you need to activate the tags within your taxonomy.

**How to use Taxonomy in MISP**

**Filtering the distribution of events among MISP instances**

Applying rules for distribution based on tags:

**MISP Taxonomies - tools**

- `machinetag.py` is a parsing tool to dump taxonomies expressed in Machine Tags (Triple Tags) and list all valid tags from a specific taxonomy.

```bash
% cd tools
% python machinetag.py
admiralty-scale:source-reliability="a"
admiralty-scale:source-reliability="b"
admiralty-scale:source-reliability="c"
admiralty-scale:source-reliability="d"
admiralty-scale:source-reliability="e"
admiralty-scale:source-reliability="f"
admiralty-scale:information-credibility="1"
admiralty-scale:information-credibility="2"
admiralty-scale:information-credibility="3"
admiralty-scale:information-credibility="4"
admiralty-scale:information-credibility="5"
admiralty-scale:information-credibility="6"
...
```

- `PyTaxonomies` - Python module to use the MISP Taxonomies

**Other use cases using MISP taxonomies**

Tags can be used to:

- Set events for further processing by external tools (e.g. VirusTotal auto-expansion using Viper).
- Ensure a classification manager classes the events before release (e.g. release of information from air-gapped/classified networks).
- Enrich IDS export with tags to fit your NIDS deployment.

**Future functionalities related to MISP taxonomies**

- Sighting support (thanks to NCSC-NL) is integrated in MISP allowing to auto expire IOC based on user detection.
- Adjusting taxonomies (adding/removing tags) based on their score or visibility via sighting.
- Simple taxonomy editors to help non-technical users to create their taxonomies.
• Filtering mechanisms in MISP to rename or replace taxonomies/tags at pull and push synchronisation.
• More public taxonomies to be included
Galaxies

Galaxies in MISP are a method used to express a large object called 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 vocabularies available in MISP galaxy but those can be overwritten, replaced or updated as you wish. Vocabularies are from existing standards (like STIX, Veris, ATT&CK, MISP and so on) or custom ones you only use for your organization.

Existing clusters and vocabularies can be used as-is or as a template. MISP distribution can be applied to each cluster to permit a limited or broader distribution scheme.

The objective is to have a common set of clusters for organizations starting analysis but that can be expanded to localized information (which is not shared) or additional information (that can be shared).

MISP galaxy is available on Github.

Managing Galaxies in MISP

[warning] You need to have a specific role to manage Galaxies on a MISP instance.

Galaxies management is accessed using the Galaxies link on the top menu.
A list with all the galaxies existing on the server will appear.

### Galaxies

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>Preventive Measure</td>
<td>1</td>
<td>Preventive measures based on the ransomware document overview as published in <a href="https://docs.google.com/spreadsheet/1TP23hucAbObfK165u6s5bfWICExW4JY89Hmcf5g/pubhtml">https://docs.google.com/spreadsheet/1TP23hucAbObfK165u6s5bfWICExW4JY89Hmcf5g/pubhtml</a>. The preventive measures are quite generic and can fit any standard Windows infrastructure and their security measures.</td>
</tr>
<tr>
<td>22</td>
<td>Ransomware</td>
<td>1</td>
<td>Ransomware galaxy based on <a href="https://docs.google.com/spreadsheet/1TP23hucAbObfK165u6s5bfWICExW4JY89Hmcf5g/pubhtml">https://docs.google.com/spreadsheet/1TP23hucAbObfK165u6s5bfWICExW4JY89Hmcf5g/pubhtml</a></td>
</tr>
<tr>
<td>21</td>
<td>TDS</td>
<td>2</td>
<td>TDS is a list of Traffic Direction System used by adversaries.</td>
</tr>
<tr>
<td>20</td>
<td>Exploit-Kit</td>
<td>2</td>
<td>Exploit-Kit is an enumeration of some exploitation kits used by adversaries. The list includes document, browser and router exploit kits. It's not meant to be totally exhaustive but aim at covering the most seen in the past 5 years</td>
</tr>
<tr>
<td>19</td>
<td>Tool</td>
<td>1</td>
<td>Threat actors tools is an enumeration of tools used by adversaries. The list includes malware but also common software regularly used by the adversaries.</td>
</tr>
<tr>
<td>18</td>
<td>Threat Actor</td>
<td>1</td>
<td>Threat actors are characteristics of malicious actors (or adversaries) representing a cyber attack threat including presumed intent and historically observed behaviour.</td>
</tr>
<tr>
<td>17</td>
<td>Microsoft Activity</td>
<td>1</td>
<td>Activity groups as described by Microsoft.</td>
</tr>
</tbody>
</table>

Page 1 of 1, showing 7 records out of 7 total, starting on record 1, ending on 7
Each galaxy can be explored using the **View** icon at the end of the line.

### Tool galaxy

<table>
<thead>
<tr>
<th>Galaxy ID</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Tool</td>
</tr>
<tr>
<td>Uuid</td>
<td>9b0b637777c6984e1-e497-3726565180b8</td>
</tr>
<tr>
<td>Description</td>
<td>Threat actors tools is an enumeration of tools used by adversaries. The list includes malware but also common software regularly used by the adversaries.</td>
</tr>
<tr>
<td>Version</td>
<td>1</td>
</tr>
</tbody>
</table>

#### Value Synonyms Activity #Events Description Actions

<table>
<thead>
<tr>
<th>Value</th>
<th>Synonyms</th>
<th>Activity</th>
<th>#Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>EyePyramid Malware</td>
<td>0</td>
<td>Two Italians referred to as the “Occisionero brothers” have been arrested and accused of using malware and a carefully-prepared spear-phishing scheme to spy on high-profile politicians and businessmen. This case has been called “EyePyramid”, which we first discussed last week.</td>
<td><img src="image" alt="Link" /></td>
</tr>
<tr>
<td>Adelind</td>
<td>AlienSpy</td>
<td>1</td>
<td>Adelind is a backdoor written purely in Java that targets system supporting the Java runtime environment. Commands that can be used, among other things, to display messages on the system, open URLs, update the malware, download/execute files, and download/load plugins. A significant amount of additional functionality can be provided through downloadable plugins, including such things as remote control options and shell command execution.</td>
</tr>
<tr>
<td></td>
<td>Frutas</td>
<td></td>
<td><img src="image" alt="Link" /></td>
</tr>
<tr>
<td></td>
<td>Unrecom</td>
<td></td>
<td><img src="image" alt="Link" /></td>
</tr>
<tr>
<td></td>
<td>Socket</td>
<td></td>
<td><img src="image" alt="Link" /></td>
</tr>
<tr>
<td></td>
<td>jSocket</td>
<td></td>
<td><img src="image" alt="Link" /></td>
</tr>
<tr>
<td></td>
<td>Jrat</td>
<td></td>
<td><img src="image" alt="Link" /></td>
</tr>
<tr>
<td></td>
<td>Backdoor:Java/Adelind</td>
<td></td>
<td><img src="image" alt="Link" /></td>
</tr>
</tbody>
</table>
Here the metadata of the selected galaxy is shown. You also see a table with each available value as well as some complementary data such as a description of the value or the activity (MISP Sightings), that is to say the evolution of the use of each value.

Galaxies can be reimported from the submodules by clicking the “Update Galaxies” link on either the galaxies list or while browsing a specific galaxy. A popup will appear to confirm the reimportation.
All galaxies will always be updated, even while browsing a specific galaxy.

**Adding a custom Galaxy repository in MISP (WiP - notFunctional)**

Fork the misp-galaxy repository to your github account.

Once you have forked the repo you can do the following, assuming you have followed the Standard MISP Install.

```
cd /var/www/MISP/app/files/
sudo rm -rf misp-galaxy
# Replace the following line with your fork
sudo -u www-data git clone https://github.com/SteveClement/misp-galaxy.git
```

Once this is done double check if you can still see the Galaxies in the Web UI.

[warning] This will impact the UI "Update MISP" functionality in administration. Your git head might get detached in your misp-galaxy repo.

**Adding a new Galaxy**

**Context**

A galaxy is designed to provide more info than a tag. It comes in two formats: regular or matrix-shape. In a tag, you can only display one label and one color. In a galaxy, you can display:

- name
- synonymous
- description
- categories (for matrix-galaxies)

**Directory structure**

Galaxies are represented by two json files stored in:

```
/var/www/MISP/app/files/misp-galaxy/galaxies/mygalaxy.json
/var/www/MISP/app/files/misp-galaxy/clusters/mygalaxy.json
```

The /galaxies file contains metatdatas and galaxy structure. The /clusters file contains actual data.

**The galaxy managyment GUI**
<table>
<thead>
<tr>
<th>ID</th>
<th>Icon</th>
<th>Name</th>
<th>Version</th>
<th>Namespace</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>406</td>
<td>🕵️‍♂️</td>
<td>Social Engineering - Dark Patterns</td>
<td>1</td>
<td>miap</td>
<td>List of vendors selling surveillance technologies including malware, interception devices or computer exploitation services</td>
</tr>
<tr>
<td>405</td>
<td>🕵️‍♂️</td>
<td>Social Engineering - Dark Patterns</td>
<td>1</td>
<td>miap</td>
<td>List of vendors selling surveillance technologies including malware, interception devices or computer exploitation services</td>
</tr>
<tr>
<td>404</td>
<td>🕵️‍♂️</td>
<td>Misinformation Pattern</td>
<td>4</td>
<td>miap</td>
<td>MITT Tactic</td>
</tr>
<tr>
<td>403</td>
<td>🕵️‍♂️</td>
<td>Regions UN 1449</td>
<td>2</td>
<td>miap</td>
<td>Regions based on UN 1449</td>
</tr>
<tr>
<td>402</td>
<td>🕵️‍♂️</td>
<td>Target Information</td>
<td>1</td>
<td>miap</td>
<td>Description of targets of threat actors</td>
</tr>
<tr>
<td>401</td>
<td>🕵️‍♂️</td>
<td>365-exchange-techniques</td>
<td>1</td>
<td>miap</td>
<td>365-exchange-techniques - Office365Exchange related techniques by @john_lew1C</td>
</tr>
<tr>
<td>400</td>
<td>🕵️‍♂️</td>
<td>CIFS/Share</td>
<td>1</td>
<td>miap</td>
<td>CIFS/Share - Principles of BITM E ATT&amp;CK in the fraud domain</td>
</tr>
<tr>
<td>399</td>
<td>🕵️‍♂️</td>
<td>Election guidelines</td>
<td>1</td>
<td>miap</td>
<td>Universal Development and Security Guidelines as Applicable to Election Technology</td>
</tr>
<tr>
<td>398</td>
<td>🕵️‍♂️</td>
<td>Wikipedia</td>
<td>1</td>
<td>miap</td>
<td>Malware galaxy based on Wikipedia archive</td>
</tr>
<tr>
<td>397</td>
<td>🕵️‍♂️</td>
<td>Backdoor</td>
<td>1</td>
<td>miap</td>
<td>Malware Backdoor galaxy</td>
</tr>
<tr>
<td>396</td>
<td>🕵️‍♂️</td>
<td>Stealer</td>
<td>1</td>
<td>miap</td>
<td>Malware Stealer galaxy</td>
</tr>
<tr>
<td>395</td>
<td>🕵️‍♂️</td>
<td>Mobile ATT&amp;CK - Relationship</td>
<td>4</td>
<td>miap</td>
<td>Malware Relationship</td>
</tr>
<tr>
<td>394</td>
<td>🕵️‍♂️</td>
<td>Mobile ATT&amp;CK - Malware</td>
<td>5</td>
<td>miap</td>
<td>Name of ATT&amp;CK software</td>
</tr>
</tbody>
</table>
In this window, you will be able to check all your galaxies and if your newly created ones are OK.

The galaxy file

The galaxy file provides the framework for the data stored in the cluster file. For example:

```json
{
    "description": "attck4fraud - Principles of MITRE ATT&CK in the fraud domain",
    "icon": "map",
    "kill_chain_order": {
        "fraud-tactics": [
            "Initiation",
            "Target Compromise",
            "Perform Fraud",
            "Obtain Fraudulent Assets",
            "Assets Transfer",
            "Monetisation"
        ]
    },
    "name": "attck4fraud",
    "namespace": "misp",
    "type": "financial-fraud",
    "uuid": "cc8c8ae9-aec2-42c0-9939-f4f82b051836",
    "version": 1
}
```
- **description**: generalities about the galaxy (1)
- **icon**: the icon used in the MISP interface (2)
- **name**: the name of the galaxy (3)
- **namespace**: the namespace where is stored the galaxy. Namespace are used to regroup similar galaxies (4)
- **type**: IMPORTANT field, it MUST match the galaxy and cluster files name to actually chain both files together (5)
- **uuid**: as any MISP object, it has a uuid. IMPORTANT, it MUST be repeated in the uuid property of the cluster file (6)
- **version**: as usual in MISP, versioning, especially to force update (7)
- **kill_chain_order**: a special and optionnal field: it will be used if you want to create a matrix-galaxy. In this field, you insert a named table (fraud-tactics in the example above) containing the categories labels of you data. They will be used then in the cluster file (8)


### The cluster file

The cluster file provides the actual data of the galaxy. For example (Attck4fraud):

```json
{
  "authors": [
    "Francesco Bigarella"
  ],
  "category": "guidelines",
  "description": "attck4fraud - Principles of MITRE ATT&CK in the fraud domain",
  "name": "attck4fraud",
  "source": "Open Sources",
  "type": "financial-fraud",
  "uuid": "cc0c8ae9-aec2-42c6-9939-f4f82b851836",
  "values": [
    {
      "description": "In the context of ATT&CK for Fraud, phishing is described as the sending of fraudulent emails to a large audience in order to obtain sensitive information (PII, credentials, payment information). Phishing is never targeted to a specific individual or organisation. Phishing tries to create a sense of urgency or curiosity in order to capture the victim.",
      "meta": {
        "detection": "Email sender is spoofed; Email sender belongs to a domain recently created; Presence of typos or poor grammar in the email text; The request in the mail is unsolicited and creates urgency; No recollection of the subject or the sender of the phishing email; Request for credentials; Presence of a suspicious URL or attachment."
      },
      "examples": [
        "Phishing messages were sent to Amazon users posing as the Amazon customer support",
        "Fake Apple invoices were sent to Apple App Store customers in order to obtain their Apple ID credentials"
      ],
      "external_id": "FT1801",
      "kill_chain": [
        "fraud-tactics:Initiation"
      ],
      "mitigation": "Implementation of DKIM and SPF authentication to detected spoofed email senders; anti-phishing solutions."
    },
    {
      "refs": [
        "https://blog.malwarebytes.com/cybercrime/2015/02/amazon-notice-ticket-number-phish-seeks-card-detai".
        "https://tools.ietf.org/html/draft-dulaunoy-misp-galaxy-format-06#page-9"
      ]
    }
  ],
  "version": 3
}
```
Galaxies

```json
{
    "authors": ["Francesco Bigarella"],
    "category": "guidelines",
    "description": "attackfraud - Principles of MITRE ATTACK in the fraud domain",
    "name": "attackfraud",
    "source": "Open Sources",
    "type": "financial-fraud",
    "uid": "cc03cb39-ae2c-42c6-9939-4f8f2b651836",
    "values": [{
        "description": "In the context of ATT&CK for Fraud, phishing is described as the sending of fraudulent emails to a large audience.
        "meta": {
            "detection": "Email sender is spoofed; email sender belongs to a domain recently created; Presence of typos or poor grammar; Phishing messages were sent to Amazon users posing as the Amazon customer support.",
            "external_id": "FT1001",
            "kill_chain": ["Fraud-tactics:Initiation"
        },
        "mitigation": "Implementation of DKIM and SPF authentication to detect spoofed email senders; anti-phishing solutions.",
        "refs": ["https://blog.malwarebytes.com/cybercrime/2015/02/amazon-notification-number-phish-seeks-card-details/",
    }
}
```
- **authors**: descriptive field (1)
- **category**: descriptive field (2)
- **description**: descriptive field (3)
- **name**: same as in /galaxy file, used in the Matrix display (4)
- **source**: descriptive field (5)
- **type**: IMPORTANT, this field MUST match the /galaxy and /cluster files names AND the type field in the /galaxy file name -5 in above paragraph- (6)
- **uuid**: IMPORTANT, this field MUST match the /galaxy uuid field -6 in above paragraph- (7)
- **values**: a table containing the actual values (8)
- **data files**: fields used to describe single data are detailed here: https://tools.ietf.org/html/draft-dulaunoy-misp-galaxy-format-06#page-9 (9)
- **kill_chain**: IMPORTANT, provide the column of the Matrix where the data will be displayed: (10)
  - **arg1**: MUST match /galaxy file's killchain arg (_fraud-tactics in the example)
  - **arg2**: name of the column of the data (Initiation in the example)
- **version**: same as for galaxies

More details on /cluster fields can be found here: https://tools.ietf.org/html/draft-dulaunoy-misp-galaxy-format-06#page-9

**Implementation**

- Once your files are ready, ALWAYS submit them in a json validator such as: https://jsonformatter.curiousconcept.com/. Do it before putting them into your instance, your sanity is at stake.

- Copy/paste your files in both folders (/galaxies and /clusters)

- Go to Galaxies/List galaxies and click on Update galaxies

- Your new galaxy should be displayed on the screen with the others

---

**Galaxies**

<table>
<thead>
<tr>
<th>Name</th>
<th>Churn Assessment</th>
<th>Galaxies</th>
<th>Input Filters</th>
<th>Galaxy Address</th>
<th>Sys Address</th>
<th>Internet</th>
<th>Admin</th>
<th>MISP</th>
<th>Actions</th>
</tr>
</thead>
</table>

**shadowrun galaxy**

- **Galaxy ID**: 41
- **Name**: shadowrun
- **Namespace**: MGP
- **ID**: T/20504-0/11-0-120-10j5-198749325311
- **Description**: Ur (Shadowrun intel galaxy)
- **Version**: 1

<table>
<thead>
<tr>
<th>Value</th>
<th>Sensitive</th>
<th>Activity</th>
<th>#Results</th>
<th>Description</th>
<th>Filter</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Page 1 of 1, showing 0 records out of 0 total, starting on record 0, ending on 0

- [MISP](#) - [Admin](#)

---

Galaxies

272
Your galaxy is available in the events for selecting in the right namespace.
Troubleshooting

- **The galaxy does not update, galaxy is empty**
  - Check json validation
  - Remove commas on last items of any {} or []
  - Update version of files
  - Check files names
  - Delete the galaxy in the GUI and update

- **Matrix is not displayed**
  - Check the kill_chain_order array in the /galaxies json
  - Check the chaining

Example

We will create a galaxy from scratch. To demonstrate MISP can handle any type of use-case, we will not work on malware but on Shadowrun pen and paper RPG. In this RPG, 2060's large megacorporations launch paramilitary actions against each other. They can belong to 3 main categories (ranked by international standards):

- **AAA**: extraterritorial corporation and seating at the top-10 council;
- **AA**: only extraterritorial companies;
- **A**: nation-scale corporation.

A corporation can act in several fields:

- energy
- IT
- biotechnology
- cybertechnology (body enhancement)

It can work on several continent:

- Europe;
- Asia;
- Africa;
- Oceania;
- America.

All these context elements are enough to build a galaxy.

**Simple galaxy**

- the galaxy file: galaxies/shadowrun.json

```json
{
  "description": "My Shadowrun test galaxy",
  "icon": "user-secret",
  "name": "shadowrun",
  "namespace": "RPG",
  "type": "shadowrun",
  "uuid": "7a956b4d-613c-4ce8-b5d6-19974682ae8",
  "version": 1
}
```

Keep the uuid and type, it will be necessary later.
- Check your json
- Click on update and see your work:

![Galaxies screenshot]

<table>
<thead>
<tr>
<th>Galaxy ID</th>
<th>A1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Shadowrun</td>
</tr>
<tr>
<td>Namespace</td>
<td>RPG</td>
</tr>
<tr>
<td>Hash</td>
<td>7e9500a4-e113-4c38-b595-99e74a09202b</td>
</tr>
<tr>
<td>Description</td>
<td>My Shadowrun test galaxy</td>
</tr>
<tr>
<td>Version</td>
<td>1</td>
</tr>
</tbody>
</table>

Page 1 of 1, showing 1 records out of 1 total, starting on record 1, ending on 1
the cluster file: clusters/shadowrun.json

```json
{
  "authors": ["myself"],
  "category": "RPG",
  "description": "Shadowrun galaxy",
  "name": "shadowrun corporations",
  "source": "Internal",
  "type": "shadowrun",
  "uuid": "7a956b4d-613c-4c88-b5d6-19974682aea8",
  "values": [
    {
      "description": "extraterritorial corporation and seating at the top-10 council."
    },
    {
      "description": "only extraterritorial compagnies."
    },
    {
      "description": "nation-scale corporation."
    },
    {
      "description": "energy sector: exploitation, refining, selling"
    }
  ]
}
```
"electricity",
"gas",
"bio"
}
},
"uuid": "293e7e5c-51a8-411f-9b47-d52ed62d4b78",
"value": "energy"
},
{
"description": "cybertechnology sector: manufacturing, selling and implanting modifications.",
"meta": {
"Delta clinic (for implanting)": [
"Yes",
"No"
],
"examples": [
"headware",
"bodyware",
"eyeware",
"earware",
"cyberlimbs"
]
},
"uuid": "7e962290-cba7-49ad-95c2-115575c8a9d2",
"value": "cybertechnology"
},
{
"description": "Biotechnology: biware, genetics, etc",
"meta": {
"examples": [
"bioware",
"genetics",
"biodrones",
"biocosmetics"
]
},
"uuid": "c899564c-bfe4-460f-a2ed-aae98e1355a3",
"value": "biotechnology"
},
{
"description": "IT: softwares, hardware, cybersec",
"meta": {
"examples": [
"software dev",
"hardware manufacturing",
"intrusion countermeasures"
]
},
"uuid": "16c49ba4-8a79-4f67-a98a-07c08f8a2d",
"value": "IT"
},
{
"description": "Europe",
"meta": {
"examples": [
"France",
"Belgium",
"Luxembourg",
"Germany",
"Italy"
]
},
"uuid": "8e745c22-9b14-4334-887a-0008eda58f75",
"value": "Europe"
},
{
"description": "Asia",
"meta": {
"examples": [null]}}
"China",
"Japan",
"Thailand"
]
},
"uuid": "95d4ff78-42f8-4fe8-bb63-af2c7e500ec8",
"value": "Asia"
},
{
"description": "Russia and former USSR",
"meta": {
"examples": [
"Russia",
"kazakhstan"
]
},
"uuid": "87a3ac88-6ff0-45eb-b26e-e8e8af392563",
"value": "Russia"
},
{
"description": "Africa",
"meta": {
"examples": [
"Nigeria",
"Malia",
"Algeria"
]
},
"uuid": "aba7f8b7-fcb4-4bfa-b8d4-b896314f5e3d",
"value": "Africa"
},
{
"description": "Oceania",
"meta": {
"examples": [
"Asutralia",
"Polynesia"
]
},
"uuid": "ae28839b-b90f-48d9-b809-acda8864ff4e",
"value": "Oceania"
},
{
"description": "America",
"meta": {
"examples": [
"UCAS",
"CAS",
"Pueblo Corporate C0uncil",
"AZtlan"
]
},
"uuid": "d41c6222-4d10-43e9-9a8e-47d86eaf0e7",
"value": "America"
}
"version": 3
}

IMPORTANT:
- the "uuid": "7a956b4d-613c-4c08-b5d6-19974682aea8"," is the same in both files
- the cluster filename is the same as the "type" field in the galaxy file
- CHECK YOUR JSON (https://jsonformatter.curiousconcept.com/) AND SAVE YOUR SANITY!

We check the thing by clicking on the update button in the galaxy GUI:
<table>
<thead>
<tr>
<th>Value</th>
<th>Synonym</th>
<th>Activity</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>0</td>
<td>total: 1 corporation</td>
</tr>
<tr>
<td>AA</td>
<td></td>
<td>0</td>
<td>only extraterrestrial companies</td>
</tr>
<tr>
<td>AAA</td>
<td></td>
<td>0</td>
<td>extraterrestrial corporation and existing in the top-10 council</td>
</tr>
</tbody>
</table>
We can test our work on the MISP GUI:

Attribute warning: This event doesn’t contain any attributes. It’s strongly advised to populate the event with attributes.
Galaxies

shadowrun

Add
shadowrun galaxy

Galaxy ID: 62
Name: Shadowrun
Namespace: RPQ
Uuid: 799f6b4d-613c-4c08-b5d5-199746828e68
Description: My Shadowrun test galaxy
Version: 2

<table>
<thead>
<tr>
<th>Value</th>
<th>Synonyms</th>
<th>Activity</th>
<th>#Events</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>0</td>
<td>nation-scale corporation.</td>
<td></td>
</tr>
<tr>
<td>AA</td>
<td>AA</td>
<td>0</td>
<td>only extraterritorial companies.</td>
<td></td>
</tr>
<tr>
<td>AAA</td>
<td>AAA</td>
<td>0</td>
<td>extraterritorial corporation and seating at the top-10 council.</td>
<td></td>
</tr>
<tr>
<td>Africa</td>
<td>Africa</td>
<td>0</td>
<td>Africa</td>
<td></td>
</tr>
<tr>
<td>America</td>
<td>America</td>
<td>0</td>
<td>America</td>
<td></td>
</tr>
<tr>
<td>Asia</td>
<td>Asia</td>
<td>0</td>
<td>Asia</td>
<td></td>
</tr>
<tr>
<td>Europe</td>
<td>Europe</td>
<td>0</td>
<td>Europe</td>
<td></td>
</tr>
<tr>
<td>IT</td>
<td>IT</td>
<td>0</td>
<td>IT: softwares, hardware, cybersecurity</td>
<td></td>
</tr>
<tr>
<td>Oceania</td>
<td>Oceania</td>
<td>0</td>
<td>Oceania</td>
<td></td>
</tr>
<tr>
<td>Russia</td>
<td>Russia</td>
<td>0</td>
<td>Russia and former USSR</td>
<td></td>
</tr>
<tr>
<td>biotechnology</td>
<td>biotechnology</td>
<td>0</td>
<td>Biotechnology: bioware, genetics, etc.</td>
<td></td>
</tr>
<tr>
<td>cybertechnology</td>
<td>cybertechnology</td>
<td>0</td>
<td>Cybertechnology sector: manufacturing, setting and implementing modifications.</td>
<td></td>
</tr>
<tr>
<td>energy</td>
<td>energy</td>
<td>0</td>
<td>energy sector: exploitation, refining, selling</td>
<td></td>
</tr>
</tbody>
</table>

Page 1 of 1, showing 13 records out of 15 total, starting on record 1, ending on 13
Remark: we created a simple galaxy. We will later see how to create a Matrix-shaped one.

**Matrix-shaped galaxy**

To create a matrix-shaped galaxy, a new field is added:

- **kill_chain** for the /galaxy json
- **kill_chain_order** for the /cluster json

In the galaxy json, categories are listed:

```json
"kill_chain":[
   "killchain_name":[
      "category_1",
      "category_2",
      "category_3"
   ]
]
```

The final galaxy file:

```json
{
   "description": "My Shadowrun test matrix galaxy",
   "icon": "user-secret",
   "kill_chain_order": {
      "shadowrun": {
         "ranking",
         "sector",
         "area"
      },
      "name": "shadowrun_matrix",
      "namespace": "RPG",
      "type": "shadowrun",
      "uuid": "1b013b10-5c6e-11ea-8881-0800275bbff6",
      "version": 1
   }
}
```

In the cluster json, reference to the categories are done:

```json
"values": [
   {
      "description": "",
      "meta": {
         "kill_chain": [
            "killchain_name:category_1"
         ]
   },
   ]
```

The final cluster file:

```json
{
   "authors": ["myself"],
   "category": "RPG",
   "description": "Shadowrun matrix galaxy",
   "name": "shadowrun corporations",
   "source": "Internal",
   "type": "shadowrun",
   "uuid": "1b013b10-5c6e-11ea-8881-0800275bbff6",
   "values": [
      
   ]
}
```
"description": "extraterritorial corporation and seating at the top-10 council.",
"meta": {
  "kill_chain": [
    "shadowrun:ranking"
  ],
  "Corporate council seat": "Yes",
  "examples": [
    "Renraku",
    "Shiawase",
    "Aztechnology",
    "Ares Macrotechnologies",
    "Saeder Krupps"
  ],
  "uuid": "43e1b900-5a03-11ea-9ad1-080027cbfd66",
  "value": "AAA"
},

{ "description": "only extraterritorial compagnies.",
"meta": {
  "kill_chain": [
    "shadowrun:ranking"
  ],
  "Corporate council seat": "No",
  "examples": [
    "Shibata",
    "Monobe",
    "Zeta Impchem",
    "ESUS"
  ],
  "uuid": "7aad2dd4-5a03-11ea-ad69-080027cbfd66",
  "value": "AA"
},

{ "description": "nation-scale corporation.",
"meta": {
  "kill_chain": [
    "shadowrun:ranking"
  ],
  "Corporate council seat": "No",
  "examples": [
    "Genom",
    "KSAF",
    "Seretech",
    "Infocore",
    "Microdek (ex-Microsoft)",
    "Tan Tien"
  ],
  "uuid": "50c0d622-5c67-11ea-bd4b-0800275bbff6",
  "value": "A"
},

{ "description": "energy sector: exploitation, refining, selling",
"meta": {
  "kill_chain": [
    "shadowrun:sector"
  ],
  "examples": [
    "Saeder Krupps"
  ],
  "subsectors": [
    "petroleum",
    "electricity",
    "gas",
    "bio"
  ]
}}
The final result:

<table>
<thead>
<tr>
<th>Ranking</th>
<th>Sector</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>IT</td>
<td>Africa</td>
</tr>
<tr>
<td>A</td>
<td>biotechnology</td>
<td>America</td>
</tr>
<tr>
<td>AAA</td>
<td>cyber technology</td>
<td>Asia</td>
</tr>
<tr>
<td>AAA</td>
<td>energy</td>
<td>Europe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Oceania</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Russia</td>
</tr>
</tbody>
</table>
Done! Eventually!

**Dependencies**

To create your own Galaxies the following tools are needed to run the validation scripts.

- jsonschema (>=v2.4)
- jq
- moreutils (sponge)

On a Debian flavoured distribution you can potentially do this:

```
sudo apt install jq moreutils python3-jsonschema
sudo wget -O /usr/local/bin/jsonschema https://gist.githubusercontent.com/SteveClement/e6ac00e153e9657913000216fc7c7ef/raw/c273ace06ad338d6096d2c84a8a6e215a268eaa11/jsonschema
sudo chmod +x /usr/local/bin/jsonschema  # This will only work with jsonschema >2.4 (before no CLI interface was available)
```

**Create a fork**

To add your custom Galaxy it is preferable to fork the misp-galaxy repository. See above for details.

**Understanding directory structure**

**Removing a Galaxy to better understand the add**

Let's start with removing a single Galaxy.

```
cd /var/www/MISP/app/files/misp-galaxy
sudo -u www-data rm galaxies/android.json
sudo -u www-data rm clusters/android.json
sudo -u www-data /var/www/MISP/app/Console/cake Admin updateGalaxies force
```

After this you will have removed the android Galaxy Cluster.

**Using Galaxies in MISP Events - Example**

For this example, we will try to add a cluster to an existing event. This cluster contains information about threat actor known as Sneaky Panda.
### Test Event

<table>
<thead>
<tr>
<th>Event ID</th>
<th>790</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uuid</td>
<td>580b20cf-2d26-4b1c-bbe4-404a950d210f</td>
</tr>
<tr>
<td>Org</td>
<td>CIRCL</td>
</tr>
<tr>
<td>Owner org</td>
<td>CIRCL</td>
</tr>
<tr>
<td>Contributors</td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td>[redacted]</td>
</tr>
<tr>
<td>Tags</td>
<td>admirally-scale:information-credibility=&quot;1&quot;</td>
</tr>
<tr>
<td>Date</td>
<td>2016-10-22</td>
</tr>
<tr>
<td>Threat Level</td>
<td>High</td>
</tr>
<tr>
<td>Analysts</td>
<td>Initial</td>
</tr>
<tr>
<td>Distribution</td>
<td>Your organisation only</td>
</tr>
<tr>
<td>Info</td>
<td>Test Event</td>
</tr>
<tr>
<td>Published</td>
<td>No</td>
</tr>
<tr>
<td>Sightings</td>
<td>1 (1)</td>
</tr>
</tbody>
</table>

**Galaxies**

- Add new cluster
Here on the event view, we notice a blue frame under the metadatas with the title "Galaxies" and a button "Add new cluster". Let's click on the latter to begin.

<table>
<thead>
<tr>
<th>Select Cluster Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Galaxies</td>
</tr>
<tr>
<td>Galaxy: Threat Actor</td>
</tr>
<tr>
<td>Galaxy: Tool</td>
</tr>
<tr>
<td>Cancel</td>
</tr>
</tbody>
</table>
A popup will appear proposing to explore a particular galaxy or all at the same time. Here, as we know we want to as a threat actor, we will choose the second option and scroll to find Sneaky Panda (We are courageous, aren't we?).

<table>
<thead>
<tr>
<th>Select Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>search clusters...</td>
</tr>
<tr>
<td>Sandworm</td>
</tr>
<tr>
<td>ScarCruft</td>
</tr>
<tr>
<td>Scarlet Mimic</td>
</tr>
<tr>
<td>Shark Spider</td>
</tr>
<tr>
<td>Shell Crew</td>
</tr>
<tr>
<td>Silent Chollina</td>
</tr>
<tr>
<td>Sofacy</td>
</tr>
<tr>
<td>Spicy Panda</td>
</tr>
<tr>
<td>Stalker Panda</td>
</tr>
<tr>
<td>Stealth Falcon</td>
</tr>
<tr>
<td>Stone Panda</td>
</tr>
</tbody>
</table>

Cancel
Wait. No Sneaky Panda? Hm that's strange. Or maybe it is only registred as a alias. Let's have a look! To do so we will use the search field which stay on top of the list. So what do we get? Beijing Group, is it an alias of our threat actor.
Pointing the cursor on it will give us the answer.

![Select Cluster]

Galaxies
We have a match. So we select it and here we go.
Clicking on the magnifying glass next to Threat actor redirects to the list of all threat actors. Clicking on the magnifying glass next to Beijing Group redirects us to a page about this group. Clicking on the addition symbol on the left of Beijing Group extends the module.

**Available Galaxies**

**Clusters**

- **Android** - Android malware galaxy based on multiple open sources.
- **Backdoor** - A list of backdoor malware.
- **Banker** - A list of banker malware.
- **Botnet** - botnet galaxy
- **Branded vulnerability** - List of known vulnerabilities and attacks with a branding
- **Cert eu govssector** - Cert EU GovSector
- **Exploit kit** - Exploit-Kit is an enumeration of some exploitation kits used by adversaries. The list includes document, browser and router exploit kits. It's not meant to be totally exhaustive but aim at covering the most seen in the past 5 years
- **Malpedia** - Malware galaxy cluster based on Malpedia.
- **Microsoft activity group** - Activity groups as described by Microsoft
- **Mitre attack pattern** - ATT&CK tactic
- **Mitre course of action** - ATT&CK Mitigation
- **Mitre enterprise attack attack pattern** - ATT&CK tactic
- **Mitre enterprise attack course of action** - ATT&CK Mitigation
- **Mitre enterprise attack intrusion set** - Name of ATT&CK Group
- **Mitre enterprise attack malware** - Name of ATT&CK software
- **Mitre enterprise attack tool** - Name of ATT&CK software
- **Mitre intrusion set** - Name of ATT&CK Group
- **Mitre malware** - Name of ATT&CK software
- **Mitre mobile attack attack pattern** - ATT&CK tactic
- **Mitre mobile attack course of action** - ATT&CK Mitigation
- **Mitre mobile attack intrusion set** - Name of ATT&CK Group
- **Mitre mobile attack malware** - Name of ATT&CK software
- **Mitre mobile attack tool** - Name of ATT&CK software
- **Mitre pre attack attack pattern** - ATT&CK tactic
- **Mitre pre attack intrusion set** - Name of ATT&CK Group
- **Mitre tool** - Name of ATT&CK software
Preventive measure - Preventive measures based on the ransomware document overview as published in https://docs.google.com/spreadsheets/d/1TWS238xacAto-fLKh1n5uTsdijWdCEsGIM0Y0Hvmc5g/pubhtml#. The preventive measures are quite generic and can fit any standard Windows infrastructure and their security measures.

Ransomware - Ransomware galaxy based on https://docs.google.com/spreadsheets/d/1TWS238xacAto-fLKh1n5uTsdijWdCEsGIM0Y0Hvmc5g/pubhtml and http://pastebin.com/raw/GHgpWjar

Rat - remote administration tool or remote access tool (RAT), also called sometimes remote access trojan, is a piece of software or programming that allows a remote "operator" to control a system as if they have physical access to that system.

Sector - Activity sectors

Stealer - A list of malware stealer.

Tds - TDS is a list of Traffic Direction System used by adversaries

Threat actor - Known or estimated adversary groups targeting organizations and employees. Adversary groups are regularly confused with their initial operation or campaign.

Tool - threat-actor-tools is an enumeration of tools used by adversaries. The list includes malware but also common software regularly used by the adversaries.

Vocabularies

Common

Certainty level - Certainty level of an associated element or cluster.

Sector - List of activity sectors

Threat actor type - threat actor type vocab as defined by Cert EU.

Ttp category - ttp category vocab as defined by Cert EU.

Ttp type - ttp type vocab as defined by Cert EU.

threat-actor

Cert eu motive - Motive vocab as defined by Cert EU.

Intended effect - The IntendedEffectVocab is the default STIX vocabulary for expressing the intended effect of a threat actor

Motivation - The MotivationVocab is the default STIX vocabulary for expressing the motivation of a threat actor.

Planning and operational support - The PlanningAndOperationalSupportVocab is the default STIX vocabulary for expressing the planning and operational support functions available to a threat actor.

Sophistication - The ThreatActorSophisticationVocab enumeration is used to define the default STIX vocabulary for expressing the subjective level of sophistication of a threat actor.

Type - The ThreatActorTypeVocab enumeration is used to define the default STIX vocabulary for expressing the subjective type of a threat actor.
Sightings

Basically, sighting is a system allowing people to react on attributes on an event. It was originally designed to provide an easy method for user to tell when they see a given attribute, giving it more credibility.

Now sightings have been improved to also provide a method to signal false positives, but also to give an expiration date for some attributes.

Explanation

As said before, Sighting is a way for a user to say that they have seen or notice an attribute and confirm its validity. An attribute can been spotted several times by the same user, that is why a single user can use sighting several times on a single attribute.

Sometimes, some attributes can be considered as false positives, even if the false positive list do not detect them (for instance, if the IDS flag is set to false) so they can also be notified. As well as concerning sighting, the same user can signal a single attribute as a false positive several times.

It also happens that some attributes are only valid a certain time (for instance, in case of a phishing campaign that is assumed to be up for only one week). In this case, people can also assign an expiration date to an attribute, but this time, there can be only one valid expiration date per organisation.

Using sightings on an event (GUI)

Sighting is applied to every attribute, under the column “Sightings”, easily identifiable with its colored number. This column shows three icons and three values.
These three values show respectively:

- The number of true positives detected with the attribute, in green. Malicious activity as described in the event.
- The number of times the attribute has been marked as false positive, in red. Non-malicious activity or incorrect detection.
- The number of different expiration dates that have been affected on this attribute, in orange.

Concerning the three icons:

- The first one (Thumb up) allows to add a sighting (true positive) on an attribute.
- The second one (Thumb down) allows to mark the attribute as a false positive.
- The third one (Tool) opens a popup for advanced sightings, showing sightings details and allowing different actions.

**Advanced sightings**

- The first tab, "Graph", represents a line graph showing the evolution of sightings and false positives over time.
- The second tab gives a quick view of all the sightings applied to the attribute.

<table>
<thead>
<tr>
<th>Date</th>
<th>Organisation</th>
<th>Type</th>
<th>Source</th>
<th>Event ID</th>
<th>Attribute ID</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-04-22 08:14:03</td>
<td>Select Astronomy</td>
<td>Expiration</td>
<td></td>
<td>1120</td>
<td>303395</td>
<td></td>
</tr>
<tr>
<td>2017-04-07 08:03:28</td>
<td>Select Astronomy</td>
<td>Expiration</td>
<td></td>
<td>1120</td>
<td>303395</td>
<td></td>
</tr>
<tr>
<td>2017-04-04 08:43:25</td>
<td>Select Astronomy</td>
<td>Sighting</td>
<td></td>
<td>1120</td>
<td>303395</td>
<td></td>
</tr>
<tr>
<td>2017-04-04 08:10:47</td>
<td>Select Astronomy</td>
<td>False-positive</td>
<td></td>
<td>1120</td>
<td>303395</td>
<td></td>
</tr>
<tr>
<td>2017-04-04 08:10:46</td>
<td>Select Astronomy</td>
<td>False-positive</td>
<td></td>
<td>1120</td>
<td>303395</td>
<td></td>
</tr>
<tr>
<td>2017-04-04 08:10:45</td>
<td>Select Astronomy</td>
<td>Sighting</td>
<td></td>
<td>1120</td>
<td>303395</td>
<td></td>
</tr>
<tr>
<td>2017-04-04 08:10:31</td>
<td>Select Astronomy</td>
<td>Sighting</td>
<td></td>
<td>1120</td>
<td>303395</td>
<td></td>
</tr>
<tr>
<td>2017-04-04 07:50:00</td>
<td>Select Astronomy</td>
<td>False-positive</td>
<td></td>
<td>1120</td>
<td>303395</td>
<td></td>
</tr>
<tr>
<td>2017-04-04 07:49:58</td>
<td>Select Astronomy</td>
<td>Sighting</td>
<td></td>
<td>1120</td>
<td>303395</td>
<td></td>
</tr>
</tbody>
</table>
The third tab gives a quick view of the sightings applied to the attribute by your own organisation only.

<table>
<thead>
<tr>
<th>Date</th>
<th>Organisation</th>
<th>Type</th>
<th>Source</th>
<th>Event ID</th>
<th>Attribute ID</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-04-07 08:09:28</td>
<td></td>
<td>Expiration</td>
<td>1120</td>
<td>303395</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2017-04-04 08:43:25</td>
<td></td>
<td>Sighting</td>
<td>1120</td>
<td>303595</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2017-04-04 07:50:00</td>
<td></td>
<td>False-positive</td>
<td>1120</td>
<td>303398</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2017-04-04 07:43:58</td>
<td></td>
<td>Sighting</td>
<td>1120</td>
<td>303335</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The last tab can be used to add either a sighting, mark the attribute as a false positive, or define an expiration date. You can precise both the date and time of day, as well as note a particular source where the sighting comes from.
At Event level

The total number of sightings is also visible as part of the metadata in front of the Sightings label, as well as a sparkline graph that summarize the evolution of sightings.

<table>
<thead>
<tr>
<th>Sightings</th>
<th>12 (7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td></td>
</tr>
</tbody>
</table>
Clicking on the tool will show sighting details for the whole event.
Using sightings on an event (API)

Please have a look at the automation API
MISP warninglists

MISP warninglists are lists of well-known indicators that can be associated to potential false positives, errors or mistakes. There is a Python module available to work with warninglists in a Pythonic way called PyMISPWarningLists. MISP warninglists GitHub Repo

MISP warning lists: The dilemma of false-positive

- False-positive is a common issue in threat intelligence sharing.
- It’s often a contextual issue:
  - false-positive might be different per community of users sharing information.
  - organization might have their own view on false-positive.
- Based on the success of the MISP taxonomy model, we build misp-warninglists. They are lists of well-known indicators that can be associated to potential false positives, errors or mistakes. They are Simple JSON files.
The warning lists are integrated in MISP to display an info/warning box at the event and attribute level. This can be enabled at MISP instance level. Default warning lists can be enabled or disabled like known public resolver, multicast IP addresses, hashes for empty values, rfc1918, TLDs or known google domains. The warning lists can be expanded or added in JSON locally or via pull requests (https://github.com/MISP/misp-warninglists). Warning lists can be also used for critical or core infrastructure warning, personally identifiable information...
MISP noticelist

Notice lists to inform MISP users of the legal, privacy, policy or even technical implications of using specific attributes, categories or objects. MISP noticelist GitHub Repo
## Attribute Categories vs. Types

<table>
<thead>
<tr>
<th>Category</th>
<th>Antivirus detection</th>
<th>Artifacts dropped</th>
<th>Attribution</th>
<th>External analysis</th>
<th>Financial fraud</th>
<th>Internal reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>aba-rtn</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>anonymised</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>attachment</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>authentihash</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bank-account-nr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bic</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bin</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>boolean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bro</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>btc</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>campaign-id</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>campaign-name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cc-number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>cdhash</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>chrome-extension-id</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>community-id</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>cookie</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cortex</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>counter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>country-of-residence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cpe</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dash</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>date-of-birth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>datetime</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dns-soa-email</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>domain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>domain</td>
<td>jp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>email</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-attachment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-body</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-dst</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-dst-display-name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-header</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-message-id</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-mime-boundary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-reply-to</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-src</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-src-display-name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-subject</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-thread-index</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-x-mailer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eppn</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>authentihash</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>impfuzzy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>imphash</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>md5</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>pehash</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha1</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha224</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha256</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha3-224</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha3-256</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha3-384</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha3-512</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha384</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha512</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha512/224</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha512/256</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>ssdeep</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>tlsh</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>vhash</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>first-name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>Types</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>float</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>frequent-flyer-number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gender</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gene</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>git-commit-id</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>github-organisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>github-repository</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>github-username</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hashs-md5</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hashs-server-md5</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hex</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hostname</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hostname</td>
<td>port</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>http-method</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iban</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>identity-card-number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>impfuzzy</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>imphash</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ip-dst</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ip-dst</td>
<td>port</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ip-src</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ip-src</td>
<td>port</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>issue-date-of-the-visa</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ja3-fingerprint-md5</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jabber-id</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jarm-fingerprint</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kusto-query</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>last-name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>link</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mac-address</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mac-eui-64</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>malware-sample</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>malware-type</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>md5</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>middle-name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mime-type</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>Mutability</td>
<td>Value</td>
<td>Type</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>-------</td>
<td>------</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mobile-application-id</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mutex</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>named pipe</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nationality</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>other</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>passenger-name-record-locator-number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>passport-country</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>passport-expiration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>passport-number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pattern-filename</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pattern-in-file</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pattern-in-memory</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pattern-in-traffic</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>payment-details</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pdb</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pehash</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pgp-private-key</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pgp-public-key</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>phone-number</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>place-of-birth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>place-port-of-clearance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>place-port-of-onward-foreign-destination</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>place-port-of-original-embarkation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>port</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>primary-residence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>prtn</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>redress-number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>regkey</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>regkey/value</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sha1</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sha224</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sha256</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sha3-224</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sha3-256</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sha3-256</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sha3-384</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sha3-512</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sha384</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sha512</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sha512/224</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sha512/256</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sigma</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>size-in-bytes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>snort</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>special-service-request</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ssdeep</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stix2-pattern</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>target-email</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>target-external</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>target-location</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>target-machine</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>target-org</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>target-user</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>telhash</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>text</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>threat-actor</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tlsh</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>travel-details</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>twitter-id</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>uri</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>url</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>user-agent</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vhash</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>visa-number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vulnerability</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>weakness</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>whois-creation-date</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>whois-registrant-email</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>whois-registrant-name</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>whois-registrant-org</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>whois-registrant-phone</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>Network activity</td>
<td>Other</td>
<td>Payload delivery</td>
<td>Payload installation</td>
<td>Payload type</td>
<td>Persistence mechanism</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------</td>
<td>-------</td>
<td>------------------</td>
<td>----------------------</td>
<td>--------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>AS</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>aba-rtn</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>anonymised</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>attachment</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>authentihash</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>bank-account-nr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>boolean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>bro</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>btc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>campaign-id</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>campaign-name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cc-number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cdhash</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>chrome-extension-id</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>community-id</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cookie</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>cortex</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>counter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>country-of-residence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cpe</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dash</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>date-of-birth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>datetime</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dns-soa-email</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>domain</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>domain</td>
<td>ip</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-attachment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-body</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-src</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-src-display-name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-header</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-message-id</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-mime-boundary</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-reply-to</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-src-display-name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-subject</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-thread-index</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-x-mailer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eppn</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>authentihash</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>imphash</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>imphash</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>md5</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>pehash</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha1</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha224</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha256</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha3-224</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha3-256</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha3-384</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha3-512</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha384</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha512</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha512/224</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha512/256</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>ssdeep</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>tish</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>vhash</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>first-name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>float</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>frequent-flyer-number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gender</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gene</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>git-commit-id</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>github-organisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>github-repository</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>github-username</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hash-md5</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hashsserver-md5</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hex</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>hostname</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hostname</td>
<td>port</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>http-method</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iban</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>identity-card-number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>impfuzzy</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>imphash</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ip-dst</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ip-dst</td>
<td>port</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ip-src</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ip-src</td>
<td>port</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>issue-date-of-the-visa</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ja3-fingerprint-md5</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>jabber-id</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jarm-fingerprint</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kusto-query</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>last-name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>link</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mac-address</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mac-eui-64</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>malware-sample</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>malware-type</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>md5</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>middle-name</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mime-type</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mobile-application-id</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mutex</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>named pipe</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nationality</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>other</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>passenger-name-record-locator-number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>passport-country</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>passport-expiration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>passport-number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pattern-filename</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pattern-in-file</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pattern-in-memory</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pattern-in-traffic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>payment-details</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pdb</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pehash</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pgp-private-key</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pgp-public-key</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>phone-number</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>place-of-birth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>place-port-of-clearance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>place-port-of-onward-foreign-destination</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>place-port-of-original-embarkation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>port</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>primary-residence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>prtn</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>redress-number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

317
<table>
<thead>
<tr>
<th>regkey</th>
<th></th>
<th></th>
<th></th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>regkey/value</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>sha1</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sha224</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>sha256</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>sha3-224</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>sha3-256</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>sha3-384</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>sha3-512</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>sha384</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>sha512</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>sha512/224</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>sha512/256</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>sigma</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>size-in-bytes</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>snort</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>special-service-request</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ssdeep</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>stix2-pattern</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>target-email</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>target-external</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>target-location</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>target-machine</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>target-org</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>target-user</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>teflhash</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>text</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>threat-actor</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tsh</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>travel-details</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>twitter-id</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>uri</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>url</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>user-agent</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>vhash</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>visa-number</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>Person</td>
<td>Social network</td>
<td>Support Tool</td>
<td>Targeting data</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>vulnerability</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>weakness</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>whois-creation-date</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>whois-registrant-email</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>whois-registrant-name</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>whois-registrant-org</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>whois-registrant-phone</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>whois-registrar</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>windows-scheduled-task</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>windows-service-displayname</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>windows-service-name</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x509-fingerprint-md5</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>x509-fingerprint-sha1</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>x509-fingerprint-sha256</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>xmr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>yara</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>zeek</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>Types</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>community-id</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cookie</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cortex</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>counter</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>country-of-residence</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cpe</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dash</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>date-of-birth</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>datetime</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dns-soa-email</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>domain</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>domain</td>
<td>ip</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-attachment</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-body</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-dst</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-dst-display-name</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-header</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-message-id</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-mime-boundary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-reply-to</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-src</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-src-display-name</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-subject</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-thread-index</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email-x-mailer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eppn</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>authentihash</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>impfuzzy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>imphash</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>md5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>pehash</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha224</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>sha256</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>Type</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filenameSHA3-224</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filenameSHA3-256</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filenameSHA3-384</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filenameSHA3-512</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filenameSHA384</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filenameSHA512</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filenameSHA512/224</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filenameSHA512/256</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filenameSSDeep</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filenameTLSH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filenameVHash</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FirstName</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Float</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frequent-Flyer-Number</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gender</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gene</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Git-Commit-Id</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Github-Organisation</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Github-Repository</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Github-Username</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hash-MD5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HashServer-MD5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hex</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hostname</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hostname</td>
<td>Port</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP-Method</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identity-Card-Number</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Imfuzzy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Imphash</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP-DST</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP-DST</td>
<td>Port</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP-SRC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP-SRC</td>
<td>Port</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue-Date-of-the-Visa</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JA3-Fingerprint-MD5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Categories and Types
<table>
<thead>
<tr>
<th>Category</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>jabber-id</td>
<td></td>
</tr>
<tr>
<td>jarm-fingerprint</td>
<td></td>
</tr>
<tr>
<td>kusto-query</td>
<td></td>
</tr>
<tr>
<td>last-name</td>
<td></td>
</tr>
<tr>
<td>link</td>
<td></td>
</tr>
<tr>
<td>mac-address</td>
<td></td>
</tr>
<tr>
<td>mac-eui-64</td>
<td></td>
</tr>
<tr>
<td>malware-sample</td>
<td></td>
</tr>
<tr>
<td>malware-type</td>
<td></td>
</tr>
<tr>
<td>md5</td>
<td></td>
</tr>
<tr>
<td>middle-name</td>
<td></td>
</tr>
<tr>
<td>mime-type</td>
<td></td>
</tr>
<tr>
<td>mobile-application-id</td>
<td></td>
</tr>
<tr>
<td>mutex</td>
<td></td>
</tr>
<tr>
<td>named pipe</td>
<td></td>
</tr>
<tr>
<td>nationality</td>
<td></td>
</tr>
<tr>
<td>other</td>
<td></td>
</tr>
<tr>
<td>passenger-name-record-locator-number</td>
<td></td>
</tr>
<tr>
<td>passport-country</td>
<td></td>
</tr>
<tr>
<td>passport-expiration</td>
<td></td>
</tr>
<tr>
<td>passport-number</td>
<td></td>
</tr>
<tr>
<td>pattern-filename</td>
<td></td>
</tr>
<tr>
<td>pattern-in-file</td>
<td></td>
</tr>
<tr>
<td>pattern-in-memory</td>
<td></td>
</tr>
<tr>
<td>pattern-in-traffic</td>
<td></td>
</tr>
<tr>
<td>payment-details</td>
<td></td>
</tr>
<tr>
<td>pdb</td>
<td></td>
</tr>
<tr>
<td>pehash</td>
<td></td>
</tr>
<tr>
<td>pgp-private-key</td>
<td></td>
</tr>
<tr>
<td>pgp-public-key</td>
<td></td>
</tr>
<tr>
<td>phone-number</td>
<td></td>
</tr>
<tr>
<td>place-of-birth</td>
<td></td>
</tr>
<tr>
<td>place-port-of-clearance</td>
<td></td>
</tr>
<tr>
<td>place-port-of-onward-foreign-destination</td>
<td></td>
</tr>
<tr>
<td>place-port-of-original-embarkation</td>
<td></td>
</tr>
<tr>
<td>port</td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>X</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---</td>
</tr>
<tr>
<td>primary-residence</td>
<td></td>
</tr>
<tr>
<td>prtn</td>
<td></td>
</tr>
<tr>
<td>redress-number</td>
<td>X</td>
</tr>
<tr>
<td>regkey</td>
<td></td>
</tr>
<tr>
<td>regkey/value</td>
<td></td>
</tr>
<tr>
<td>sha1</td>
<td></td>
</tr>
<tr>
<td>sha224</td>
<td></td>
</tr>
<tr>
<td>sha256</td>
<td></td>
</tr>
<tr>
<td>sha3-224</td>
<td></td>
</tr>
<tr>
<td>sha3-256</td>
<td></td>
</tr>
<tr>
<td>sha3-384</td>
<td></td>
</tr>
<tr>
<td>sha3-512</td>
<td></td>
</tr>
<tr>
<td>sha384</td>
<td></td>
</tr>
<tr>
<td>sha512</td>
<td></td>
</tr>
<tr>
<td>sha512/224</td>
<td></td>
</tr>
<tr>
<td>sha512/256</td>
<td></td>
</tr>
<tr>
<td>sigma</td>
<td></td>
</tr>
<tr>
<td>size-in-bytes</td>
<td></td>
</tr>
<tr>
<td>snort</td>
<td></td>
</tr>
<tr>
<td>special-service-request</td>
<td>X</td>
</tr>
<tr>
<td>ssdeep</td>
<td></td>
</tr>
<tr>
<td>stix2-pattern</td>
<td></td>
</tr>
<tr>
<td>target-email</td>
<td>X</td>
</tr>
<tr>
<td>target-external</td>
<td>X</td>
</tr>
<tr>
<td>target-location</td>
<td>X</td>
</tr>
<tr>
<td>target-machine</td>
<td>X</td>
</tr>
<tr>
<td>target-org</td>
<td>X</td>
</tr>
<tr>
<td>target-user</td>
<td>X</td>
</tr>
<tr>
<td>telfhash</td>
<td></td>
</tr>
<tr>
<td>text</td>
<td>X</td>
</tr>
<tr>
<td>threat-actor</td>
<td></td>
</tr>
<tr>
<td>tlsh</td>
<td></td>
</tr>
<tr>
<td>travel-details</td>
<td>X</td>
</tr>
<tr>
<td>twitter-id</td>
<td></td>
</tr>
<tr>
<td>uri</td>
<td></td>
</tr>
<tr>
<td>url</td>
<td></td>
</tr>
</tbody>
</table>
Categories

- **Antivirus detection**: All the info about how the malware is detected by the antivirus products
- **Artifacts dropped**: Any artifact (files, registry keys etc.) dropped by the malware or other modifications to the system
- **Attribution**: Identification of the group, organisation, or country behind the attack
- **External analysis**: Any other result from additional analysis of the malware like tools output
- **Financial fraud**: Financial Fraud indicators
- **Internal reference**: Reference used by the publishing party (e.g. ticket number)
- **Network activity**: Information about network traffic generated by the malware
- **Other**: Attributes that are not part of any other category or are meant to be used as a component in MISP objects in the future
- **Payload delivery**: Information about how the malware is delivered
- **Payload installation**: Info on where the malware gets installed in the system
- **Payload type**: Information about the final payload(s)
- **Persistence mechanism**: Mechanisms used by the malware to start at boot
- **Person**: A human being - natural person
- **Social network**: Social networks and platforms
- **Support Tool**: Tools supporting analysis or detection of the event
- **Targeting data**: Internal Attack Targeting and Compromise Information

Types
Categories and Types

- **AS**: Autonomous system
- **aba-rtn**: ABA routing transit number
- **anonymised**: Anonymised value - described with the anonymisation object via a relationship
- **attachment**: Attachment with external information
- **authentihash**: Authenticode executable signature hash
- **bank-account-nr**: Bank account number without any routing number
- **bic**: Bank Identifier Code Number also known as SWIFT-BIC, SWIFT code or ISO 9362 code
- **bin**: Bank Identification Number
- **boolean**: Boolean value - to be used in objects
- **bro**: An NIDS rule in the Bro rule-format
- **btc**: Bitcoin Address
- **campaign-id**: Associated campaign ID
- **campaign-name**: Associated campaign name
- **cc-number**: Credit-Card Number
- **cdhash**: An Apple Code Directory Hash, identifying a code-signed Mach-O executable file
- **chrome-extension-id**: Chrome extension id
- **comment**: Comment or description in a human language
- **community-id**: a community ID flow hashing algorithm to map multiple traffic monitors into common flow id
- **cookie**: HTTP cookie as often stored on the user web client. This can include authentication cookie or session cookie.
- **cortex**: Cortex analysis result
- **counter**: An integer counter, generally to be used in objects
- **country-of-residence**: The country of residence of a natural person
- **cpe**: Common Platform Enumeration - structured naming scheme for information technology systems, software, and packages.
- **dash**: Dash Address
- **date-of-birth**: Date of birth of a natural person (in YYYY-MM-DD format)
- **datetime**: Datetime in the ISO 8601 format
- **dns-soa-email**: RFC1035 mandates that DNS zones should have a SOA (Statement Of Authority) record that contains an email address where a PoC for the domain could be contacted. This can sometimes be used for attribution/linkage between different domains even if protected by whois privacy
- **domain**: A domain name used in the malware
- **domain|ip**: A domain name and its IP address (as found in DNS lookup) separated by a |
- **email**: An e-mail address
- **email-attachment**: File name of the email attachment.
- **email-body**: Email body
- **email-dst**: The destination email address. Used to describe the recipient when describing an e-mail.
- **email-dst-display-name**: Email destination display name
- **email-header**: Email header
- **email-message-id**: The email message ID
- **email-mime-boundary**: The email mime boundary separating parts in a multipart email
- **email-reply-to**: Email reply to header
- **email-src**: The source email address. Used to describe the sender when describing an e-mail.
- **email-src-display-name**: Email source display name
- **email-subject**: The subject of the email
- **email-thread-index**: The email thread index header
- **email-x-mailer**: Email x-mailer header
- **eppn**: eduPersonPrincipalName - eppn - the NetId of the person for the purposes of inter-institutional authentication. Should be stored in the form of user@univ.edu, where univ.edu is the name of the local security domain.
- **filename**: Filename
- **filename|authentihash**: A checksum in md5 format
- **filename|impfuzzy**: Import fuzzy hash - a fuzzy hash created based on the imports in the sample.
- **filename|imphash**: Import hash - a hash created based on the imports in the sample.
- **filename|md5**: A filename and an md5 hash separated by a |
- **filename|pehash**: A filename and a PHash hash separated by a |
- **filename|sha1**: A filename and an sha1 hash separated by a |
- **filename|sha224**: A filename and a sha-224 hash separated by a |
- **filename|sha256**: A filename and an sha256 hash separated by a |
- **filename|sha3-224**: A filename and an sha3-224 hash separated by a |
- **filename|sha3-256**: A filename and an sha3-256 hash separated by a |
- **filename|sha3-384**: A filename and an sha3-384 hash separated by a |
- **filename|sha3-512**: A filename and an sha3-512 hash separated by a |
- **filename|sha384**: A filename and a sha-384 hash separated by a |
- **filename|sha512**: A filename and a sha-512 hash separated by a |
- **filename|sha512/224**: A filename and a sha-512/224 hash separated by a |
- **filename|sha512/256**: A filename and a sha-512/256 hash separated by a |
- **filename|ssdeep**: A filename and an Ssdeep checksum in ssdeep format
- **filename|tlsh**: A filename and a Trend Micro Locality Sensitive Hash separated by a |
- **filename|vhash**: A filename and a VirusTotal hash separated by a |
- **first-name**: First name of a natural person
- **float**: A floating point value.
- **frequent-flyer-number**: The frequent flyer number of a passenger
- **gender**: The gender of a natural person (Male, Female, Other, Prefer not to say)
- **gene**: GENE - Go Evtx sigNature Engine
- **git-commit-id**: A git commit ID.
- **github-organisation**: A github organisation
- **github-repository**: A github repository
- **github-username**: A github user name
- **hash-md5**: hash is a network fingerprinting standard which can be used to identify specific Client SSH implementations. The fingerprints can be easily stored, searched and shared in the form of an MD5 fingerprint.
- **hashserver-md5**: hashserver is a network fingerprinting standard which can be used to identify specific Server SSH implementations. The fingerprints can be easily stored, searched and shared in the form of an MD5 fingerprint.
- **hex**: A value in hexadecimal format
- **hostname**: A full host/dnsname of an attacker
- **hostname|port**: Hostname and port number separated by a |
- **http-method**: HTTP method used by the malware (e.g. POST, GET, ...).
- **iban**: International Bank Account Number
- **identity-card-number**: Identity card number
- **impfuzzy**: A fuzzy hash of import table of Portable Executive format
- **imphash**: Import hash - a hash created based on the imports in the sample.
- **ip-dst**: A destination IP address of the attacker or C&C server
- **ip-dst|port**: IP destination and port number separated by a |
- **ip-src**: A source IP address of the attacker
- **ip-src|port**: IP source and port number separated by a |
- **issue-date-of-the-visa**: The date on which the visa was issued
- **ja3-fingerprint-md5**: JA3 is a method for creating SSL/TLS client fingerprints that should be easy to produce on any platform and can be easily shared for threat intelligence.
- **jabber-id**: Jabber ID
- **jarm-fingerprint**: JARM is a method for creating SSL/TLS server fingerprints.
- **kusto-query**: Kusto query - Kusto from Microsoft Azure is a service for storing and running interactive analytics over Big Data.
- **last-name**: Last name of a natural person
- **link**: Link to an external information
Categories and Types

- **mac-address**: Mac address
- **mac-eui-64**: Mac EUI-64 address
- **malware-sample**: Attachment containing encrypted malware sample
- **malware-type**: md5: A checksum in md5 format
- **middle-name**: Middle name of a natural person
- **mime-type**: A media type (also MIME type and content type) is a two-part identifier for file formats and format contents transmitted on the Internet
- **mobile-application-id**: The application id of a mobile application
- **mutex**: Mutex, use the format \BaseNamedObjects\named pipe: Named pipe, use the format \pipe\nationality: The nationality of a natural person
- **other**: Other attribute
- **passenger-name-record-locator-number**: The Passenger Name Record Locator is a key under which the reservation for a trip is stored in the system. The PNR contains, among other data, the name, flight segments and address of the passenger. It is defined by a combination of five or six letters and numbers.
- **passport-country**: The country in which the passport was issued
- **passport-expiration**: The expiration date of a passport
- **passport-number**: The passport number of a natural person
- **pattern-filename**: A pattern in the name of a file
- **pattern-in-file**: Pattern in file that identifies the malware
- **pattern-in-memory**: Pattern in memory dump that identifies the malware
- **pattern-in-traffic**: Pattern in network traffic that identifies the malware
- **payment-details**: Payment details
- **pdb**: Microsoft Program database (PDB) path information
- **pehash**: PEhash - a hash calculated based of certain pieces of a PE executable file
- **pgp-private-key**: A PGP private key
- **pgp-public-key**: A PGP public key
- **phone-number**: Telephone Number
- **place-of-birth**: Place of birth of a natural person
- **place-port-of-clearance**: The port of clearance
- **place-port-of-onward-foreign-destination**: A Port where the passenger is transiting to
- **place-port-of-original-embarkation**: The original port of embarkation
- **port**: Port number
- **primary-residence**: The primary residence of a natural person
- **prtn**: Premium-Rate Telephone Number
- **redress-number**: The Redress Control Number is the record identifier for people who apply for redress through the DHS Travel Redress Inquiry Program (DHS TRIP). DHS TRIP is for travelers who have been repeatedly identified for additional screening and who want to file an inquiry to have erroneous information corrected in DHS systems
- **regkey**: Registry key or value
- **regkey|value**: Registry value + data separated by |sha1**: A checksum in sha1 format
- **sha224**: A checksum in sha224 format
- **sha256**: A checksum in sha256 format
- **sha3-224**: A checksum in sha3-224 format
- **sha3-256**: A checksum in sha3-256 format
- **sha3-384**: A checksum in sha3-384 format
- **sha3-512**: A checksum in sha3-512 format
- **sha384**: A checksum in sha384 format
- **sha512**: A checksum in sha512 format
- **sha512/224**: A checksum in the sha-512/224 format
Categories and Types

- **sha512/256**: A checksum in the sha-512/256 format
- **sigma**: Sigma - Generic Signature Format for SIEM Systems
- **size-in-bytes**: Size expressed in bytes
- **snort**: An IDS rule in Snort rule-format
- **special-service-request**: A Special Service Request is a function to an airline to provide a particular facility for a Passenger or passengers.
- **ssdeep**: A checksum in ssdeep format
- **stix2-pattern**: STIX 2 pattern
- **target-email**: Attack Targets Email(s)
- **target-external**: External Target Organizations Affected by this Attack
- **target-location**: Attack Targets Physical Location(s)
- **target-machine**: Attack Targets Machine Name(s)
- **target-org**: Attack Targets Department or Organization(s)
- **target-user**: Attack Targets Username(s)
- **telfhash**: telfhash is symbol hash for ELF files, just like imphash is imports hash for PE files.
- **text**: Name, ID or a reference
- **threat-actor**: A string identifying the threat actor
- **tlsh**: A checksum in the Trend Micro Locality Sensitive Hash format
- **travel-details**: Travel details
- **twitter-id**: Twitter ID
- **uri**: Uniform Resource Identifier
- **url**: url
- **user-agent**: The user-agent used by the malware in the HTTP request.
- **vhash**: A VirusTotal checksum
- **visa-number**: Visa number
- **vulnerability**: A reference to the vulnerability used in the exploit
- **weakness**: A reference to the weakness used in the exploit
- **whois-creation-date**: The date of domain's creation, obtained from the WHOIS information.
- **whois-registrant-email**: The e-mail of a domain's registrant, obtained from the WHOIS information.
- **whois-registrant-name**: The name of a domain's registrant, obtained from the WHOIS information.
- **whois-registrant-org**: The org of a domain's registrant, obtained from the WHOIS information.
- **whois-registrant-phone**: The phone number of a domain's registrant, obtained from the WHOIS information.
- **whois-registrar**: The registrar of the domain, obtained from the WHOIS information.
- **windows-scheduled-task**: A scheduled task in windows
- **windows-service-displayname**: A windows service's displayname, not to be confused with the windows-service-name. This is the name that applications will generally display as the service's name in applications.
- **windows-service-name**: A windows service name. This is the name used internally by windows. Not to be confused with the windows-service-displayname.
- **x509-fingerprint-md5**: X509 fingerprint in MD5 format
- **x509-fingerprint-sha1**: X509 fingerprint in SHA-1 format
- **x509-fingerprint-sha256**: X509 fingerprint in SHA-256 format
- **xmr**: Monero Address
- **yara**: Yara signature
- **zeek**: An NIDS rule in the Zeek rule-format
Sharing / Synchronisation

MISP's core functionality is sharing where everyone can be a consumer and/or a contributor/producer.

- Quick benefit without the obligation to contribute
- Low barrier access to get acquainted to the system

Synchronisation

Concept

The following figure shows the concept how different MISP instances could tie together.
Legend:
- Operated by CIRCL
- Operated by NATO/NCIRC
- Operated by other organizations
In MISP, two ways exist to get events from remote sources:

- **Use case 1**: From another MISP server (also called MISP instance), by synchronising two MISP servers.
- **Use case 2**: From a link, by using Feeds.

The example below illustrate the **synchronisation** between two MISP servers (use case 1). An organisation B (OrgB) wants to synchronise its MISP server, called ServerB, with the MISP server of an organisation A (Org A), called ServerA. The following steps can be taken to synchronise ServerB with ServerA:
**FIGURE: Illustration of the synchronisation between two MISP servers**

- **Step 1:** Add OrgB as a local organisation on ServerA (OrgB.ServerA) using OrgB's existing UUID from their local organisation on ServerB.
- **Step 2:** Add a Sync User (syncuser@OrgB.ServerA) in the organisation OrgB.ServerA on the MISP ServerA.
- **Step 3:** Set up a sync server on MISP ServerB using the key (called Authkey) from the sync user (syncuser@OrgB.ServerA) created on MISP ServerA.

For additional information on the synchronisation process, refer to the MISP GitHub issues, for example, issue 2595.

**Adding a server**

Servers can be added by users via

https://<misp url>/servers/add
The Add Server Form has several input fields:

Add Server

Base URL

Instance name

Information about the organisation that will receive the events, typically the remote instance's host organisation.

Remote Syno Organisation Type

Local organisation

Local Organisation

CIRCL

Authkey

Push

Pull

Self Signed

Server certificate file

Client certificate file

Push rules:

Modify

Pull rules:

Modify

Submit
1. **Base URL**

The base-url to the external server you want to sync with. Example: https://foo.sig.mil.be

2. **Instance Name**

A name that will make it clear to your users what this instance is. For example: Organisation A's instance

3. **Remote Sync Organisation Type**

MISP has several organisation "pools", one for local and one for known external organisations. When adding a synchronisation connection, you need to define the host organisation of the remote instance. Select which pool you wish to pick the organisation from using this drop-down. You also have the option of adding a new organisation directly from this interface.

4. **Local/Known remote Organisation**

Choose the organisation from the selected pool that defines the host organisation on the remote side. Make sure that the remote instance is actually run by the organisation you select as this is used in an integral part of the sharing mechanism. Do not select your own organisation for this setting.

5. **Authkey**

You can find the authentication key on your profile on the external server.

6. **Push**

Allow the upload of events and their attributes. That means only Events that match the given filter will be pushed to the server.

E.g. it can limit push of events to events not being TLP:RED

1. **Pull**

Allow the download of events and their attributes from the server. That means only Events matching the given criteria will be pulled.

E.g. it can limit to NOT download Type:OSINT events.

2. **Self Signed**

Click this, if you would like to allow a connection despite the other instance using a self-signed certificate (not recommended). (server certificate file still needed)

3. **Server certificate file**

You can also upload a certificate file if the instance you are trying to connect to has its own signing authority. (*.pem)

4. **Client certificate file**

You can also upload a certificate file if the instance you are trying to connect to has its own signing authority. (*.pem)

**Test connection**

Test connection can be used to test the connection to the remote server and will give a feedback about local and remote version of MISP.

**Rules**

Rules are used to limit sharing when synchronising events and attributes, to e.g. events with a given tag, or disabling sharing for events containing a certain Tag.
Troubleshooting

If you have issues connecting to a remote servers try to do the following things:

- try to connect with your user account to the remote server, to ensure the password is still valid and that your API key is valid
- try to connect with your user account to the remote server and check your roles on the remote server
- with connection issues do a package capture to find out more
- if you have a SSL connection issue to a remote server with a signed by a CA that is not included in OS, make sure the whole certificate path is included in the path.

Sharing and distribution

The following section describes how distribution mechanisms of events and attributes work.

Distribution settings

The below five distribution settings are available for events and attributes. Descriptions of those settings can be found [here](#).

- Your organisation only
- This community only
- Connected communities
- All communities
- Sharing group

Events that are not published are only distributed/shared to the local organisations on the same MISP server/instance (within the limit of the distribution model). Only events that are **published** will be shared with remote organisations on other MISP servers via push/pull mechanisms. More details on publishing events [here](#).

Community

A community is composed of the local organisations on a MISP server and the remote organisations connected by the sync users. For more information on the concept of community, refer to an [article on MISP information sharing following ISO/IEC 27010](#), explaining the concept of community.

Specifically, communities are not reversible. Taking the example of the above figure, illustrating the synchronisation between two MISP servers, OrgB.ServerB is part of the MISP ServerA community but OrgB.ServerA is not part of MISP ServerB community.

Distribution mechanisms

The distribution level of an event is automatically decreased as it is synchronised with other MISP instances, when it was originally set to:

- Community only (to organisation only)
- Connected community (to community only)

It is not decreased when it was originally set to:

- Organisation only
- All communities
- Sharing group

[!] This rule does not apply if “Internal instance” has been checked when creating the server.
As an example, the figure below illustrates two events \( e \) and \( e' \) created by OrgA and respectively shared as “This community only” and “Connected communities” and how they propagate in an illustrative MISP set of instances synchronised with each others.
**Sharing-groups**

There is an article about sharing groups in here.

**Collaboration**

**Proposals**

Proposals can be used to propose new attribute values that can be reviewed by the event owner.

**Forums / Threats**

Forums can be used to discuss non event related topics.

Discussions can be accessed on the top "Global Actions - List Discussions"

Discussions will and can not be shared with other servers

and via URL:

```
https://<misp url>/threads/index
```
Create a new Topic

To create a new topic

https://<misp url>/posts/add

Add Post

Thread Subject
This is a test subject

I would like to talk about foo bar because...

Submit
Comment a topic

A topic can be commented by any user

https://<misp url>/threads/view/<topic id>

Comments to events

In MISP ongoing events can be commented by every user to ask free text question to events. Comments to events will not be shared with other servers
**Contact a reporter**

This feature can be used to contact the person or the organisation that the person belongs to that has created the event.

All E-Mails can be enforced to be encrypted
Receive alerts

It is possible to get alerts via encrypted mail in the following cases:

- published events by other user of the MISP instance
- events pushed to the MISP instance
- events pulled by the MISP instance

These E-Mail alerts are an opt-in feature

Edit My Profile

<table>
<thead>
<tr>
<th>Email</th>
<th>Password</th>
<th>Confirm Password</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:admin@misp.training">admin@misp.training</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Organisation</th>
<th>Role</th>
<th>Nids Sid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>admin</td>
<td>4000000</td>
</tr>
</tbody>
</table>

GPG key

[Add GPG key]

- Receive alerts when events are published
- Receive alerts from "contact reporter" requests

Submit
Recommendation

The following section will describe what is the best practice how many MISP instances that showed to be good for orgs. Of course depending on your specific requirements an architecture could be more spread or simplified.

The architecture is divided into several systems / stages beginning with:

**MISP Staging System**

This system's purpose is to be linked to all available external MISP systems that you have access to. It will download all events and do enrichment between these events.

**MISP SECOps System**

This system is the main system used by human analysts. It will it is not linked to any external MISP instance other then the Staging System.

To publish events to the community assign the right tags to match your push Rules and publish the event
External Connectors

The MISP to Microsoft Graph Security Script enables you to connect your custom threat indicators or Indicators of Compromise (IoCs) and make these available in the following Microsoft products.

Azure Sentinel

Microsoft Defender ATP

MISP to Microsoft Graph Security Script

The script provides clients with MISP instances to migrate threat indicators to the Microsoft Graph Security API.

For more information on Microsoft Graph Security API visit Microsoft Graph Security API.
For more information on Microsoft Graph visit Microsoft Graph.

Prerequisites

Before installing the sample:

- Install Python 3.x version from https://www.python.org/.
- To register your application for access to Microsoft Graph, you'll need either a Microsoft account or an Office 365 for business account. If you don't have one of these, you can create a Microsoft account for free at outlook.com.

Getting Started

After the prerequisites are installed or met, perform the following steps to use these scripts:

1. Download or clone this repository.
2. Go to directory security-api-solutions/Samples/MISP
3. Install dependencies. In the command line, run pip3 install requests requests-futures pymisp
4. To run script, go to the root directory of misp-graph-script and enter PYTHONHASHSEED=0 python3 script.py in the command line.

App Registration

To configure the sample, you'll need to register a new application in the Microsoft Application Registration Portal. Follow these steps to register a new application:

1. Sign in to the Application Registration Portal using either your personal or work or school account.
2. Choose New registration.
3. Enter an application name, and choose Register.

4. Next you’ll see the overview page for your app. Copy and save the Application Id field. You will need it later to complete the configuration process.

5. Under Certificates & secrets, choose New client secret and add a quick description. A new secret will be displayed in the Value column. Copy this password. You will need it later to complete the configuration process and it will not be shown again.

6. Under API permissions, choose Add a permission > Microsoft Graph.

7. Under Application Permissions, add the permissions/scopes required for the sample. This sample requires ThreatIndicators.ReadWrite.OwnedBy.

   Note: See the Microsoft Graph permissions reference for more information about Graph’s permission model.

8. Modify the RequestManager.py file to comment out line 121-124. (This allows the script to run without failing due to line 123 being divided by avg_speed incase it starts as 0.

9. Modify the script.py to add in config.misp_verifycert at line 13. Ensure it looks like below.

   ```python
   misp = PyMISP(config.misp_domain, config.misp_key, config.misp_verifycert)
   ```

10. Modify config.py file to add in misp_verifycert = False anywhere in the file.

As the final step in configuring the script, modify the config.py file in the root folder of your cloned repo.

Update tenant, client_id, and client_secret in config.py

```python
graph_auth = {
    'tenant': '<tenant id>',
    'client_id': '<client id>',
    'client_secret': '<client secret>',
}
```

Once changes are complete, save the config file.

### Configurations

#### Target Product

```python
targetProduct = "Azure Sentinel" or targetProduct = "Microsoft Defender ATP"
```

#### Misp Event Filter

Filters can be set in the config.py file under the "misp_event_filters" property

Below is a list of parameters that can be passed to the filter (source: [https://pymisp.readthedocs.io/modules.html]):

- values – values to search for
- not_values – values not to search for
- type_attribute – Type of attribute
- category – Category to search
- org – Org reporting the event
- tags – Tags to search for
- not_tags – Tags not to search for
- date_from – First date (Format: '2019-01-01')
date_to – Last date (Format: '2019-01-01')
last – Last published events (for example 5d or 12h or 30m)
eventid – Event ID
withAttachments – return events with or without the attachments
uuid – search by uuid
publish_timestamp – the publish timestamp (Note: Uses UNIX timestamp. Format: '1551811160')
published – return only published events (Format: True or False)

A list or a specific value can be passed to the above parameters. If a list is passed to the parameter, the filtered events are the result of the union of provided list.

This field needs to be a list that contains multiple filters. The filtered events are the result of the intersection of provided filters.

First Example of How This Field can be Configured

```python
misp_event_filters = [
    {
        "type_attribute": 'mutex'
    },
    {
        "type_attribute": 'filename|md5'
    }
]
```

An event meets this filtering criteria if the event has an attribute with attribute type of 'mutex' AND the event has an attribute with attribute type of 'filename|md5'.

Second Example of How This Field can be Configured

```python
misp_event_filters = [
    {
        "type_attribute": ['mutex', 'filename|md5']
    }
]
```

An event meets this filtering criteria if the event has an attribute with attribute type of 'mutex' OR the event has an attribute with attribute type of 'filename|md5'.

Third Example of How This Field can be Configured

```python
misp_event_filters = [
    {
        "values": 'http://www.test.com'
    }
]
```

An event meets this filtering criteria if the event has an attribute with attribute value of 'http://www.test.com'.

Fourth Example of How This Field can be Configured

```python
misp_event_filters = []
```

This gets all events.
Action

Possible action values are: alert, allow, block.

```
action = "alert"  # (This is default).
```

Passive Only

```
passiveOnly = False  # (This is default).
```

Days to Expire

This property is used to specify the amount of days the records will expire in Microsoft Graph Security API. The default value for days to expire is 30.

```
days_to_expire = 5
```

Misp Key

The Misp Auth Key is required to fetch data from your Misp instance. Configure a sync user.

```
misp_key = '<misp key>'
```

Verify Cert

This gives you the option to choose if python should validate the certificate of the misp instance. (This allows ease within testing environments)

```
misp_verifycert = False  # IT IS RECOMMENDED TO USE A VALID SSL CERT IN PRODUCTION AND CHANGE THIS TO TRUE
```

Instructions on Reading TiIndicators That Have Been Pushed

In the command line, run `python3 script.py -r`

Instructions on Seeing All Requests That Resulted in Errors

1. In the command line, run `cd logs` to go to the logs folder.
2. To print all the requests that resulted in errors to the console, simply run `cat *.error_*` in the command line.
   - To aggregate all the requests that resulted in errors to a file, run `cat *.error_* > <filename>.txt` in the command line.

Script Output

As the script runs, it prints out the request body sent to the Microsoft Graph Security API and the response from the Microsoft Graph Security API.

Every request is logged as a json file under the directory "logs". The name of the json file is the datetime of when the request is completed.

Schedule with CRONTAB
Below is a CRONTAB entry example of running the script every Sunday at 2am

0 2 Sun /home/mark/misp-graph-script/python3 script.sh

This README.md has been adapted from the README.md found in the Microsoft Graph Security API MISP sample. For most recent changes, visit Microsoft Graph Security API MISP sample. Provide your feedback on this sample by filing a GitHub request.
MISP modules

MISP modules are autonomous modules that can be used for expansion and other services in MISP. The modules are written in Python 3 following a simple API interface. The objective is to ease the extensions of MISP functionalities without modifying core components. The API is available via a simple REST API which is independent from MISP installation or configuration.

MISP modules support is included in MISP starting from version 2.4.28.

More

MISP modules GitHub Repo

Installation

Install guide on Ubuntu

Install guide RHEL/CentOS
MISP ZeroMQ

MISP includes a flexible publish-subscribe model to allow real-time integration of the MISP activities (event publication, attribute creation or removal, sighting). The MISP ZeroMQ plugin operates at global level in MISP which means standard distribution rules don't apply and every activities will be published within the ZeroMQ pub-sub channels.

MISP ZeroMQ functionality can be used for various model of integration or to extend MISP functionalities:

- real-time search of indicators into a SIEM
- automatic expansion
- dashboard activities
- logging mechanisms
- continuous indexing
- custom software or scripting

The following notification topic channels exist and can be included in the MISP ZeroMQ pub-sub:

- **misp_json** - events published
- **misp_json_attribute** - attribute updated or created
- **misp_json_sighting** - sighting added to an attribute or an event
- **misp_json_user** - user updates or creation
- **misp_json_organisation** - organisation updates or creation
- **misp_json_self** - keep-alive messages sent every minute

**MISP ZeroMQ configuration**

To enable MISP ZeroMQ, the feature must be enabled in the Plugin setting tab.

Prior to enabling it, make sure that you have the pyzmq installed by running

```
sudo pip install pyzmq
sudo pip install redis
```

If you have problems and the plugin does not start, the logfile may be helpful.

```
sudo cat /var/www/MISP/app/tmp/logs/mispzmq.error.log
```
### Server settings

<table>
<thead>
<tr>
<th>Priority</th>
<th>Setting</th>
<th>Value</th>
<th>Description</th>
<th>Error Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional</td>
<td>Plugin-Zeromq_enable</td>
<td>true</td>
<td>Enables or disables the publish-subscribe feature of MISP. Make sure that you install the requirements for the plugin to work. Note: In the installation instructions for each implementation.</td>
<td></td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin-Zeromq_port</td>
<td>30008</td>
<td>The port that the publish-subscribe will use.</td>
<td></td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin-Zeromq_creds_host</td>
<td>secure</td>
<td>Location of the Redis db used by MISP and the Python Pulsar topic. The data to be published.</td>
<td></td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin-Zeromq_creds_port</td>
<td>7710</td>
<td>The port that Redis is listening on.</td>
<td></td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin-Zeromq_creds_password</td>
<td></td>
<td>The password of the Redis.</td>
<td></td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin-Zeromq_creds_database</td>
<td></td>
<td>The database to be used for storing messages for the publish-subscribe functionality.</td>
<td></td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin-Zeromq_creds_namespace</td>
<td>mspq</td>
<td>The namespace to be used for storing messages for the publish-subscribe functionality.</td>
<td></td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin-Zeromq_event_notifications_enable</td>
<td>true</td>
<td>Enables or disables the publishing of any event creations/deletions.</td>
<td></td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin-Zeromq_object_notifications_enable</td>
<td>true</td>
<td>Enables or disables the publishing of any object creations/deletions.</td>
<td></td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin-Zeromq_object_relations_notifications_enable</td>
<td>true</td>
<td>Enables or disables the publishing of any object relations creations/deletions.</td>
<td></td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin-Zeromq_attire_notifications_enable</td>
<td>true</td>
<td>Enables or disables the publishing of any attire creations/deletions.</td>
<td></td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin-Zeromq_signing_notifications_enable</td>
<td>true</td>
<td>Enables or disables the publishing of new signing to the ZMQ publish feed.</td>
<td></td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin-Zeromq_user_notifications_enable</td>
<td>true</td>
<td>Enables or disables the publishing of new/modified users in the ZMQ publish feed.</td>
<td></td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin-Zeromq_organisation_notifications_enable</td>
<td>true</td>
<td>Enables or disables the publishing of new/modified organisations to the ZMQ publish feed.</td>
<td></td>
</tr>
</tbody>
</table>
Each notification channels can be enabled (from event publication to sightings), the MISP site admin can decide which type of message to publish.

By default, the ZMQ pub-sub channel is available to localhost only on TCP port 50000. The binding of the pub-sub channel can be updated in the configuration interface as shown above

**MISP ZeroMQ debugging and testing**

In the diagnostic section, ZeroMQ service can be started and stopped. There is a small status option to give information about the numbers of events processed by the service.

### ZeroMQ

This tool tests whether the ZeroMQ extension is installed and functional.

ZeroMQ settings: OK

- Start / Restart
- Stop
- Status
Testing with sub.py tool

A simple command line tool is included with MISP to connect to the MISP ZeroMQ channel and get the notifications:

```
python3 sub.py --help
```

```

Generic ZMQ client to gather events, attributes and sighting updates from a MISP instance

optional arguments:
  -h, --help              show this help message and exit
  -s, --stats             print regular statistics on stderr
  -p PORT, --port PORT    set TCP port of the MISP ZMQ (default: 50000)
  -r HOST, --host HOST    set host of the MISP ZMQ (default: 127.0.0.1)
  -o ONLY, --only ONLY    set filter (misp_json, misp_json_attribute or misp_json_sighting) to limit the output a specific type (default: no filter)
  -t SLEEP, --sleep SLEEP sleep time (default: 2)
```

The `sub.py` will output the JSON objects for the subscribed topic, by default, all the topic channels are dumped:

```
misp@cpeb:/var/www/MISP/tools/misp-zmq$ python3 -u sub.py  | jq .
....
{
  "uptime": 50,
  "status": "And when you're dead I will be still alive."
}
{
  "uptime": 60,
  "status": "And believe me I am still alive."
}
{
  "uptime": 70,
  "status": "I'm doing science and I'm still alive."
}
{
  "uptime": 80,
  "status": "I feel FANTASTIC and I'm still alive."
}
{
  "uptime": 90,
  "status": "While you're dying I'll be still alive."
}
```

```
"Sighting": {
  "uuid": "592d9588-fda0-490f-bf6e-4e56950d210f",
  "source": "",
  "type": "0",
  "date_sighting": 1496159624,
  "org_id": "2",
  "event_id": "8102",
  "attribute_id": "1044812"
}
```

```
"Attribute": {
  "id": "1044802",
  "value2": "",
  "value1": "1.2.3.4",
  "uuid": "592d8494-7120-4760-b5e2-4858950d210f",
  "batch_import": "0",
  "comment": "",
  "value": "1.2.3.4",
```
Notification Schemas

Each notification channel uses a slightly different JSON schema. Consult this section to identify which MISP components exist in a channel:

**misp_json - events published**

When an event is published to ZMQ (which is different from being published in MISP) the ZMQ notification will just contain the MISP event data along with all its component children. These components include:

- A list of attributes
- A list of objects, which contain their own lists of attributes
- A list of related events - added when attributes in separate events correlate
- Any galaxies that this event belongs to
- A list of tags that apply to the event

Example:

```json
{
  "Event": {
    "id": "625",
    "orgc_id": "2",
    "org_id": "1",
    "date": "2017-05-24",
    "threat_level_id": "3",
    "info": "M2M - Fwd: IMG_3428.pdf",
    "published": false,
    "uuid": "59259036-fcd0-4749-8a6c-4d8b8950d210f",
    "attribute_count": "7",
    "analysis": "1",
    "timestamp": "1585755566",
    "distribution": "3",
    "proposal_email_lock": false,
    "locked": false,
    "publish_timestamp": "1585416766",
    "sharing_group_id": "0",
    "disable_correlation": false,
    "Org": {
      "id": "1",
      "name": "MISP",
      "uuid": "56ef3277-1ad4-42f6-b90b-04e5b983b382"},
    "Orgc": {
      "id": "2",
      "name": "CIRCL",
      "uuid": "55f6ea5e-2c60-4e5b-9e64f-47a8950d210f"},
    "Attribute": [
      {
        "id": "157835",
        "type": "attachment",
        "category": "Artifacts dropped",
        "to_ids": false,
        "uuid": "59259037-1814-4669-90b1-4b8af950d210f",
        "event_id": "625",
        "distribution": "5",
        "timestamp": "1495633975",
        "comment": "IMG_3428.pdf",
```
"sharing_group_id": "0",
"deleted": false,
"disable_correlation": false,
"object_id": "0",
"object_relation": null,
"value": "tmpzuni0skf",
"ShadowAttribute": []
}
"
,"ShadowAttribute": [],
"RelatedEvent": [],
"Galaxy": [],
"Object": [
{
"id": "1",
"name": "http-request",
"meta-category": "network",
"description": "A single HTTP request header",
"template_uuid": "b4a8d163-8110-4239-bfcf-e08f3a9f8f7b",
"template_version": "1",
"event_id": "625",
"uuid": "59c0016c-0984-4779-9688-05b8c0a83832",
"timestamp": "1505755500",
"distribution": "5",
"sharing_group_id": "0",
"comment": "",
"deleted": false,
"ObjectReference": [],
"Attribute": [
{
"id": "164371",
"type": "http-method",
"category": "Network activity",
"to_ids": false,
"uuid": "59c0016c-a744-445d-ad92-05b8c0a83832",
"event_id": "625",
"distribution": "5",
"timestamp": "1505760143",
"comment": "",
"sharing_group_id": "0",
"deleted": false,
"disable_correlation": false,
"object_id": "1",
"object_relation": "method",
"value": "POST",
"ShadowAttribute": []
}
}],
"Tag": [{"id": "2", "name": "tlp:white", "colour": "#ffffff", "exportable": true, "hide_tag": false}]
}
misp_json_attribute - attribute updated or created

The attributes appear to have the most diversity depending on the action applied to them.

When an attribute gets created, just the attribute gets sent out via ZMQ. Its parent event id is sent inside the attribute JSON, but there is no extra event metadata like there is when an attribute is deleted or modified.

Create Example:

```json
{
    "Attribute": {
        "to_ids": "1",
        "timestamp": 1505235275,
        "distribution": "5",
        "deleted": "0",
        "disable_correlation": "0",
        "event_id": "625",
        "category": "Network activity",
        "type": "domain",
        "value": "microsoft.net",
        "comment": "",
        "batch_import": "0",
        "uuid": "59b8114b-1c80-4149-be3a-03e9c8a38332",
        "sharing_group_id": 0,
        "value1": "microsoft.net",
        "value2": "",
        "id": "164363"
    }
}
```

Edited attribute notifications send metadata about their parent events and information about the attribute's sharing group, attribute-level tags, and sightings data. It's important to note that only the new value of the edited attribute is sent along the ZMQ channel. In order to diff the new and old values, you'd have to have a copy of the old attribute value stored somewhere and can use the attribute's `uuid` key (which never changes) to correlate the new and old values.

Edit Example:

```json
// microsoft.net --> microsoft.com
{
    "Attribute": {
        "id": "164363",
        "event_id": "625",
        "category": "Network activity",
        "type": "domain",
        "value1": "microsoft.com",
        "value2": "",
        "to_ids": "1",
        "uuid": "59b8114b-1c80-4149-be3a-03e9c8a38332",
        "timestamp": 1505235283,
        "distribution": "5",
        "sharing_group_id": 0,
        "comment": "",
        "deleted": false,
        "disable_correlation": false,
        "value": "microsoft.com",
        "batch_import": "0"
    },
    "Event": {
        "id": "625",
        "org_id": "1"
    }
}
```
When an attribute gets deleted, the `deleted` key gets set to `1`, and the attribute's event metadata gets sent alongside it.

Delete Example:

```
{
    "Attribute": {
        "id": "164362",
        "event_id": "625",
        "category": "Network activity",
        "type": "domain",
        "value1": "microsoft.com",
        "value2": "",
        "to_ids": true,
        "uuid": "59b81121-f4b4-4ed3-aa43-03eac8a83832",
        "timestamp": 1505235262,
        "distribution": "9",
        "sharing_group_id": 0,
        "comment": "",
        "deleted": 1,
        "disable_correlation": false,
        "value": "microsoft.net"
    },
    "Event": {
        "id": "625",
        "org_id": "1",
        "date": "2017-05-24",
        "info": "M2M - Fwd: IMG_3428.pdf",
        "user_id": "1",
        "uuid": "59259036-fcd0-4749-8a6c-4d88950d210f",
        "published": false,
        "analysis": "1"
    }
}
```
misp_json_sighting - sighting added to an attribute or an event

The message sent for sightings is fairly simple, with the type of sighting (0 = Addition, 1 = False Positive), the date (in seconds-since-epoch format), the id of the attribute it applies to, and the id of the attribute's parent event.

Addition Example:

```json
{
  "Sighting": {
    "type": "0",
    "attribute_id": "164373",
    "event_id": "625",
    "org_id": "1",
    "date_sighting": 1505767537,
    "source": "",
    "uuid": "59c03071-f480-4311-a710-03edc0a83832",
    "id": "1"
  }
}
```

False Positive Example:

```json
{
  "Sighting": {
    "type": "1",
    "attribute_id": "164373",
    "event_id": "625",
    "org_id": "1",
    "date_sighting": 1505767543,
    "source": "",
    "uuid": "59c03077-d560-4a8b-b841-05b8c0a83832",
    "id": "2"
  }
}
```

misp_json_user - user updates or creation

An update is sent through ZMQ when users log in. There are actually two messages in this - both being fairly sparse. The `current_login` message just contains who logged in and what time (in seconds-since-epoch format) it happened. The `last_login` message contains who just logged in, what time the login occurred (technically the date the record was modified, but it's modified when the user logs in, so it appears to be interchangeable in this case), and what time the user last logged in.

Login Example:

```json
{
  "User": {
    "id": "1",
    "last_login": "19900846766",
  }
}
```
When a user gets created, all of the information about the user (id, email, base64 encoded GnuPG key, role, etc.) gets sent along ZMQ. If this information is modified, the same JSON will be sent along the ZMQ channel, with updated values. For example, if the below user is disabled, the same JSON will be sent, but the disabled key will be set to "1".

User Creation and User Edit Example:

```json
{
   "User": {
      "server_id": 0,
      "autoalert": "1",
      "invited_by": "1",
      "nids_sid": 5976699,
      "termsaccepted": 0,
      "role_id": "3",
      "change_pw": 1,
      "contactalert": "1",
      "disabled": "0",
      "current_login": "0",
      "last_login": "0",
      "force_logout": "0",
      "email": "user@testemail.com",
      "enable_password": "0",
      "org_id": "1",
      "authkey": "__<redacted>__",
      "gpgkey": "__<redacted>__",
      "notify": "1",
      "date_created": 10800068160,
      "date_modified": 10800068160,
      "newsread": 0,
      "certif_public": "",
      "id": "4"
   }
}
```

misp_json_organisation - organisation updates or creation

Org notifications are sent when Orgs are updated and created, but not deleted. They are generally the same, except the fields created_by and date_created are present when an Org is created.

Creation Example:

```json
{
   "Organisation": {
      "created_by": "1",
      "local": "1",
      "name": "test",
      "uuid": "59c0367d-fe8c-42a4-9db2-03ecc8a3832",
      "description": "Test",
      "nationality": "Not specified",
      "sector": "",
      "type": "",
      "contacts": "",
      "logo": {
```
misp_json_self - keep-alive messages sent every minute

Only really useful to ensure the ZMQ server is running. And for a bit of humor

```
{
    "status": "I'm doing science and I'm still alive.",
    "uptime": 9170
}
```

Tips for Building a Subscriber

1. **misp_json_attribute** notifications are sent when attributes are created, deleted, and edited
   - Check the `deleted` key to identify if an attribute has been created or deleted
   - If an attribute has been edited, the new value will be sent out via ZMQ, but the `uuid` key will remain the same. Use this to determine if an attribute has existed before or not

2. Some compound attribute types have component types that don't exist outside of them
   - For example, MISP doesn't have a single `ip` attribute type except in the `domain|ip` type
   - If you're going to split up and resubmit these attributes, you may have to modify these component types so MISP will recognize them (e.g. `domain|ip` -> `domain, ip-dst`)
MISP and Internationalization (i18n)

Requirements

Please read the following CakePHP documentation about i18n & l10n.

Add one .md per translation effort

Please add a file à la: ja_JP.md (Japanese_Japan) or it_CH.md (Italian_Switzerland), in which you briefly describe what the current status of your translation effort is and what has been translated and which parts might be gotchas. This would also be a good place to quickly explain what your language is about, like whether most technical terms are a translation from the original, an adaptation from the English word or perhaps you just mostly use English terms.

Style

Please follow whatever is the purest and most intelligible form of written language in the native tongue being translated.

Formatting

It is important to use correct formatting. This is wrong:

```php
<p><?php echo __('Are you sure you want to delete Proposal #') . $id . '?'; ?></p>
```

You want to have ultimate flexibility and that line should look more like this:

```php
<p><?php echo __('Are you sure you want to delete Proposal #{$id}?');?></p>
```

In the above example we use an alternative notation of the format string in PHP. Using the above, the generated po-template file (default.pot) will have the name of the to-be-translated variable in the "msgid" part of the file. Which is easier to read then a non descriptive %s and allows the translator to have context on how the phrase is used in MISP.

In case you have HTML-Tags, move them out of the sentence, out of the php code if possible:

```php
<p><?php echo __('Are you sure you want to:<br>Delete Proposal #{$id}?'); ?></p>
```

Issues

Some times it might be impossible to translate some phrases. Or you notice a certain bad formatting, or segmentation of sentences. In that case, please either open an Issue on Github

Quirks
Lines like this:

```php
echo $this->Form->button('Submit', array('class' => 'btn btn-primary'));$
```

Should be prepared as such:

```php
echo $this->Form->button(__('Submit'), array('class' => 'btn btn-primary'));
```

Or another case:

```php
echo $this->Form->input('sharing_group_id', array(
    'options' => array($sharingGroups),
    'label' => __('Sharing Group'),
));
```

To:

```php
echo $this->Form->input('sharing_group_id', array(
    'options' => array($sharingGroups),
    'label' => __('Sharing Group'),
));
```

**Let us know!**

Are you planning to do a translation or localization? Please open a ticket on the [issue system](#). This will allow us and others to track what is being worked on. You can keep it very light, as all the details should be in your markdown in misp-book.

**Reach out to the community**

Want to chat with other MISP contributors? Make sure to join our [MISP Gitter channel](#).
Frequently Asked Questions

General questions
- Where can I get support?
- What are the hardware requirements?
- How to monitor MISP?

Specific questions
- Can I configure MISP encrypted notification emails to contain more information in the subject?
- How can I restart the workers?
- How can I redirect HTTP to HTTPS?
- When I try to access my new installation, I am redirected to localhost:8443 and get an error.
- How can I define the default sharing level?
- How can I add an organisation logo and/or footer logo?
- All workers are starting correctly except schdlr. How can I fix this?
- How can I import data directly from PDF reports?
- I am having trouble updating beyond version 2.4.50 (stuck loading any page beyond the login), what can I do? (-what-can-i-do?)
- I have many failed jobs when doing email notification. What should I do?
- Upgrading from MISP 2.4.65 to MISP 2.4.66 - Unable to merge due to the Composer file.
- I have issues with pushing events
- I have many users or API accesses, what's the best PHP session handler?
- Is there TAXII support?
- Wipe MISP data - Remove all data
- Constantly acknowledging my self-signed certificate drives me nuts
- How can I change the theme?
- How can I deal with a MISP instance that has pulled in feeds over and over into new events, generating hundreds of GBs of junk correlations, rendering the instance unusable?
- I have a long list of events that I want to delete via the API, do I really have to loop through each and issue a delete to /events/delete?
- I can no longer log in. How do I reset the admin password?

Usage questions
- How can I see all the deleted events in a MISP instance?

Permission issues
- RHEL/CentOS
- Redis Connection problems
- RHEL/CentOS SELinux debug
- Clearing the audit logs

When to update MISP?
- How to switch from tagged releases and back?

Update MISP fails
- What can go wrong if I update MISP?
- Error: pathspec 'app/composer.json' did not match any file(s) known to git-known-to-git)
- MISP modules "Connection refused"

Uninstalling MISP

Updating PyMISP to incorporate newer versions of the MISP object templates
- How to disable freetext/custom/user-created tags and only allow certain tags
- How to enable the csv import module?
- Why do I see 'The request has been black-holed' when I submit forms?
- Importing large feeds creates PHP Fatal error
- I deleted the admin user by mistake
FAQ

- config.php is not writeable
- How to debug misp-dashboard
- How to update object templates?
- What to do if my REST client is throwing SSL errors when trying to query my MISP instance?
- What to do if my REST client cannot reach the host, despite me being able to issue requests using Curl / Postman / etc.?
- How would one set up a sharing group with a remote org, where we only share a mutual community instance (i.e. we both have sync users on that instance). On our local instance, they exist as a remote org (from events that have synced from their instance via our shared community instance). -on-our-local-instance,-they-exist-as-a-remote-org-(from-events-that-have-synced-from-their-instance-via-our-shared-community-instance).)
- Is it possible to propose objects to an event?
- How to use the enforceWarninglist parameter in REST search?
- Column not found issue ~ Symptoms
- WatchList Customization
  - How to create a customized WatchList.
- How to upgrade PHP on RHEL/CentOS?
  - Example: Upgrade from PHP 7.2 to 7.3 on CentOS 7 ~ Enable repository ~ Install packages ~ Install required PEAR-modules ~ PHP configuration ~ Switch to PHP 7.3 ~ Disable/enable services
- How to add a galaxy to an event via PyMISP
- Updating PHP from 7.2 to 7.4.5 on Ubuntu 18.04
  - Installation
  - Verification of php 7.2 to 7.4
  - What are the required steps after a MISP installation to have a properly running instance?

Frequently Asked Questions

The following page hosts some frequently asked questions as noticed in our issues and gitter channels.

General questions

Where can I get support?

If you have feature requests or you found a bug you can open a ticket on MISP's GitHub repository issue tracker.

If you want to discuss something related to MISP or want help from the MISP community, join the appropriate MISP Gitter channel:

- MISP Developer Room Dev discussions
- MISP Support Room OMGoo! My MISP doesn't work discussions
- MISP Sharing Room Threat Intelligence Sharing discussions
- misp-cloud Room Using MISP in the clouds discussions

What are the hardware requirements?

From a hardware perspective, MISP's requirements are quite humble, a web server with 2+ cores and 8-16 GB of memory should be plenty, though more is always better of course. A lot of it depends on the data set and the number of users you are dealing with.

We recommend a standard LAMP stack on top of Ubuntu >18.04 LTS. For details on the exact dependencies please refer to the installation guide as well as the requirements for the MISP modules.
During a Hackathon a small tool called MISP-Sizer was conceived. It will give you a very rough idea on what requirements are if you have a bigger installation. source-code is here

How to monitor MISP?

Currently there are 2 documented ways to monitor MISP.

Either with MUNIN -> misp-monitor for instructions. Or OpenNMS -> Instructions here

Specific questions

Can I configure MISP encrypted notification emails to contain more information in the subject?

The setting 'MISP.extended_alert_subject' allows you to have an extended subject. \! Beware if you’re using encryption: the subject will not be encrypted. Be aware that you might leak some sensitive information this way. Below is an example how the two subject types look like. First with the option disabled, then with the option enabled.

| Event 7 - Low - TLP Amber                  |
| Event 8 - OSINT - Dissecting XXX... - Low - TLP Amber |

(Source: Getting started with MISP)

How can I restart the workers?

The workers can be restarted from the web interface:

```
administration -> server settings -> workers -> restart all
```

You can also follow the manual process below.

If you are on Ubuntu / Debian based systems:

```
sudo su -l www-data -s /bin/bash -c "bash /var/www/MISP/app/Console/worker/start.sh"
```

If you are on RHEL / Fedora based systems:

```
su -s /bin/bash apache -c 'bash /var/www/MISP/app/Console/worker/start.sh'
```

How can I redirect HTTP to HTTPS?

```
<VirtualHost *:80>
  ServerAdmin misp@misp.misp
  ServerName misp.misp.misp
  ServerAlias misp-int.misp.misp
  Redirect permanent / https://misp.misp.misp
  LogLevel warn
  ErrorLog /var/log/apache2/misp.local_error.log
  CustomLog /var/log/apache2/misp.local_access.log combined
  ServerSignature Off
</VirtualHost>
```
When I try to access my new installation, I am redirected to localhost:8443 and get an error.

By default, MISP runs on a local instance and is setup for local access upon installation. This allows you to setup security and customizations before making it available elsewhere. If you would like to access the MISP instance from a remote host (including another VM host/client), assign an IP to the MISP host and point your browser accordingly. Upon login, you may get the “localhost:8443” redirection. Change that piece of the URL back to the IP assigned to the MISP host (or associated DNS name) and refresh the browser. Once in, go to Administration - Server Settings and Maintenance - MISP settings. You can change the top two items to your MISP IP or DNS name and the redirect will start using that address instead of ‘localhost’.

How can I define the default sharing level?

MISP allows you to define the group of people with whom you want to share your threat data. If you do not set it to your preferred default then it’s likely that at one given moment you’ll make an error and share your intel with the wrong group. Defining the sharing level is done with the setting default_event_distribution in the configuration file. There are three levels:

0 : Your organisation only (default)  
1 : This community only  
2 : Connected communities  
3 : All communities

You can set a similar configuration setting for the attributes. The setting default_attribute_distribution has the same values as default_event_distribution. Additionally it has the value event which allows the attribute to get the setting from the event to which it belongs.

Source: Getting started with MISP

How can I add an organisation logo and/or footer logo?

MISP can be made more appealing to the eye by adding some graphics.
As Org.- or Site-admin navigate to Administration -> List organisations and edit the corresponding organization. With this editor you will be able to update the logo.

Other ways to achieve this, would be:

Set your organisation logo by adding an image (.png) that has the same name as your organisation in the directory /var/www/MISP/app/webroot/img/orgs/.

Yet another way of doing this is by logging into your MISP instance with Admin rights, navigate to Administration -> Server Settings, tab -> Manage files.

You can add a footer logo. Add an image to the directory /var/www/MISP/app/webroot/img/custom/ and define the footer logo in the config file (config.php) or in Administration -> Server Settings... -> MISP settings (search for: "footer_logo") point to the location on-disk of the image.

Partial source: Getting started with MISP

All workers are starting correctly except schdlr. How can I fix this?

This can happen if the FQDN of the server hosting the instance has changed. A way to fix this is to flush temporary data stored in redis. This can be done by logging in redis, for example when logging in with redis-cli, and issuing a flushall command.

How can I import data directly from PDF reports?

/!\ This section needs review, verification and eventual amendments to make sure it works.

You can use a generic script called IOC parser (https://github.com/armbues/ioc_parser) or use a script published by Palo Alto to convert IOC parser output to a MISP event (https://github.com/PaloAltoNetworks-BD/report_to_misp/). You have also the option to select all the text and paste it in the free-text import form.

Another option is the new OCR import module that can be used via the import modules. You will need to install the OCR software tesseract.

I am having trouble updating beyond version 2.4.50 (stuck loading any page beyond the login), what can I do?

/!\ This applies to an earlier version of MISP, do not randomly try this fix on valuable data. By all means try it on a test-machine and report back if your problem was solved by this.

This is most likely due to the fact that MISP did not clean up expired sessions prior to version 2.4.51 automatically and relied on a site-admin occasionally cleaning it up using the button found on the diagnostics page. Once you upgrade to 2.4.51, MISP will try to cull the table with each page load by a site-admin, which in some cases if the table has grown to extreme sizes it will get stuck on. To resolve the issue, log into mysql:

```
mysql -u [misp-db-user-name] -p [misp-db-name];
```

and execute the following commands:

```
DROP cake_sessions; CREATE TABLE IF NOT EXISTS cake_sessions ( id varchar(255) COLLATE utf8_bin NOT NULL DEFAULT '', data text COLLATE utf8_bin NOT NULL, expires int(11) NOT NULL, PRIMARY KEY (id), INDEX expires (expires)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

After this everything should work and the session table will be trimmed each time a site-admin loads a page.

I have many failed jobs when doing email notification. What should I do?
This is most probably due to some encryption failing for some users. We strongly advise to review the current PGP keys and to ensure that they keys are not expired or perhaps not supported anymore (weak keys). The keys can be reviewed at the following location in MISP:

https://<YOUR MISP URL>/users/verifyGPG

**Upgrading from MISP 2.4.65 to MISP 2.4.66 - Unable to merge due to the Composer file.**

In MISP 2.4.66, Composer is included by default to avoid the risk of downloading a rogue PHP Composer version (if the composer repository is compromised or MiTM are performed) via the download and php execution. But when upgrading (via a git pull), the git merge process might complain about the composer phar file still being there. You can safely remove that file and git pull origin 2.4 again.

**I have issues with pushing events**

- What does the 'Connection test' for the specific server report? (Sync Actions -> List Servers)
- Is the event you assume to push/pull ready to be published?
- Is the distribution level set not too restrictive?
- Have you enabled push in the servers config you want to push to?
- Do you have any limitations to the push rules e.g. limited to a certain TLP Level tag or other?
- What is written in your job log?

https://jobs/index

Have a look at: /var/www/MISP/app/tmp/logs and /var/log/apache2/misp (or the relevant apache log folder of the instance in cause)

**I have many users or API accesses, what’s the best PHP session handler?**

We strongly recommend production-level MISP installations to rely on PHP session in Redis. As Redis is already part of a standard MISP setup, we recommend to enable the redis session handling.

To configure the redis session handling in PHP, edit:

```php
session.save_handler = redis
session.save_path = "tcp://127.0.0.1:6379"
```

**Is there TAXII support?**

A TAXII 1 implementation can be found at https://github.com/MISP/MISP-Taxii-Server. This is mostly a TAXII server hooked up to MISP, meant to receive STIX files to its in box and uploading them to MISP. There is also an experimental feature to push MISP events to the TAXII server when they're published - that's in scripts/push_published_to_taxii.py. It seems to work, but may occasionally re-upload duplicate events to MISP.

TAXII 2 is provided in the future once the specification, which is at time of writing in draft, reaches a stable form.

**Wipe MISP data - Remove all data**
If you need to start from scratch with your MISP database and remove all data you can use the `misp-wipe` script provided in the `tools/` folder.

**Constantly acknowledging my self-signed certificate drives me nuts**

You want to add it in 2 places: Your browser(s) and your OS.

The following steps can be performed on the CLI to install the Certificate:

```bash
sudo mkdir -m 0755 /usr/local/share/ca-certificates/MISP
sudo cp /etc/ssl/private/misp.local.crt /usr/local/share/ca-certificates/MISP
sudo chmod 0644 /usr/local/share/ca-certificates/MISP/misp.local.crt
sudo update-ca-certificates
```

For the Chrome Browser:

1. Visit: "Advanced Settings" -> chrome://settings/?search=Manage+certificates
2. Scroll down to: Manage Certificates (click)
3. Select: "Authorities"
4. Click: "Import"
5. Browse to your .crt file and import it.
6. On the next screen tick: "Trust this certificate for identifying websites"
7. Done, enjoy the new gained quality of life

Note: Chrome might expect a Subject Alternative Name make sure you created your certificate with `-extension san`.

To allow insecure localhost connections enable this option: chrome://flags/#allow-insecure-localhost

Sources: CLI and Chrome/Chrome insecure localhost

For the Firefox Browser

**How can I change the theme?**

MISP uses `bootstrap.css` the specific CSS file can be found on a typical MISP install at `/var/www/MISP/app/webroot/css/bootstrap.css`. You can customize this for your own needs. There are also pre-made bootstrap themes which you can use as-is or build upon.

Before making any changes, confirm the version of bootstrap currently used by running `head -5 /var/www/MISP/app/webroot/css/bootstrap.css`. You can find themes on sites like Bootswatch.

To replace the current theme with a theme you found on bootsplash, run: `wget https://bootswatch.com/2/readable/bootstrap.css -O /var/www/MISP/app/webroot/css/bootstrap.css`, replacing the URL as needed.

Some bootswatch themes applied on MISP:

- [https://i.imgur.com/usONTLk.png](https://i.imgur.com/usONTLk.png)
- [https://i.imgur.com/5XMjB7o.png](https://i.imgur.com/5XMjB7o.png)
- [https://i.imgur.com/Sgc57VU.png](https://i.imgur.com/Sgc57VU.png)
- [https://i.imgur.com/4AJCPgf.png](https://i.imgur.com/4AJCPgf.png)
- [https://i.imgur.com/JuMGm8U.png](https://i.imgur.com/JuMGm8U.png)
- [https://i.imgur.com/v1WufxW.png](https://i.imgur.com/v1WufxW.png)
How can I deal with a MISP instance that has pulled in feeds over and over into new events, generating hundreds of GBs of junk correlations, rendering the instance unusable?

**Step 1:** ensure that all your CSV/freetext source_format feeds are using the fixed event setting. If you want to make sure this is the case, you can run this SQL query instead of doing it manually:

```
UPDATE feeds SET fixed_event = 1 WHERE source_format="csv" OR source_format="freetext";
```

**Step 2:** purge all of your correlations (this will make the next steps much faster), for which you have two methods at your disposal:

- either go to your administration -> server settings -> MISP tab and set MISP.completely_disable_correlation to true
- via MYSQL run TRUNCATE correlations;

**Step 3:** purge all of your feed data that have been pulled into multiple events. The easiest way of doing this: check which feeds are enabled (ignore misp source format feeds, they are not causing issues) and note down the IDs. Afterwards, use the CLI cleanup tool to remove all the feed events:

```
/var/www/MISP/app/Console/cake Admin purgeFeedEvents [user_id] [feed_id]
```

Execute this for each feed that you had enabled, replacing user_id with your admin user's ID and feed_id with the individual feed IDs on your list.

**Step 4:** recorrelate your data, depending on which method you've used in Step 2 you have two options:

- either go to your Administration -> Server Settings... -> MISP... tab and set MISP.completely_disable_correlation to false
- recorrelate your current data-set via the recorrelate attributes tool on /pages/display/administration

I have a long list of events that I want to delete via the API, do I really have to loop through each and issue a delete to /events/delete?

No, the delete action also accepts a list of IDs when it comes to bulk event deletions.

Simply POST your ID list to /events/delete in the following format:

```
{
  "id": [1,3,5,7,9]
}
```

I can no longer log in. How do I reset the admin password?

You can reset the password via the console. See Issue #1160

```
/var/www/MISP/app/Console/cake Password [email] [password]
```

Usage questions

How can I see all the deleted events in a MISP instance?

You can use the logging system for this, to see all deleted events, simply go to Audit -> Search Logs and use the following parameters:

```
model: Event
```
This will list all event deletions. To find out more about what a particular deleted event was, simply grab the ID from the above search results and search for:

```
model: Event
action: add
model_id: <Event ID retrieved from the listing of all event deletions>
```

To do the same via the API, first search for the deletions:

```
POST request:
url: https://url.of.your.misp/logs/index
headers:
  Authorization: <your_api_key>
  Accept: application/json
  Content-type: application/json
Body:
{
  "model": "Event",
  "action": "delete"
}
```

Then find the individual event's metadata that was deleted

```
POST request:
url: https://url.of.your.misp/logs/index
headers:
  Authorization: <your_api_key>
  Accept: application/json
  Content-type: application/json
Body:
{
  "model": "Event",
  "action": "add",
  "model_id": "<Event ID retrieved from the query before>"
}
```

### Permission issues

If you have any permission issues, please set the permissions to something sane first.

#### RHEL/CentOS

There are a plethora of issues that might arise when using SELinux when it comes to permissions. First, please familiarize yourself with the basics of SELinux. RedHat has a comprehensive [SELINUX USER'S AND ADMINISTRATOR'S GUIDE](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/selinux_user_s_and_administrator_s_guide/).

For file system permissions, refer to the install guide first. Another way to see what SELinux might not be happy about is to use `ausearch`. This assumes Audit is enabled.

```bash
# Just php-fpm
sudo ausearch -c 'php-fpm' -m AVC
# All messages
sudo ausearch -m AVC
```

### Redis Connection problems
If you have the following in error.log

```
2019-05-08 10:16:05 Error: [RedisException] Permission denied
Request URL: /events/view/1
Stack Trace:
#0 /var/mysql/MISP/app/Model/AppModel.php(1776): Redis->connect('127.0.0.1', 6379)
#1 /var/mysql/MISP/app/Model/Feed.php(329): AppModel->setupRedis()
#2 /var/mysql/MISP/app/Model/Event.php(2873): Feed->attachFeedCorrelations(Array, Array, Array, false)
#3 /var/mysql/MISP/app/Controller/EventsController.php(1547): Event->fetchEvent(Array, Array)
#4 [internal function]: EventsController->view('1')
#5 /var/mysql/MISP/app/Lib/cakephp/lib/Cake/Controller/Controller.php(499): ReflectionMethod->invokeArgs(Object(EventsController), Array)
#6 /var/mysql/MISP/app/Lib/cakephp/lib/Cake/Routing/Dispatcher.php(193): Controller->invokeAction(Object(CakeRequest), Array)
#7 /var/mysql/MISP/app/Lib/cakephp/lib/Cake/Routing/Dispatcher.php(167): Dispatcher->_invoke(Object(EventsController), Object(CakeRequest))
#8 /var/mysql/MISP/app/webroot/index.php(92): Dispatcher->dispatch(Object(CakeRequest), Object(CakeResponse))
#9 [main]
```

This means that apache/php-fpm cannot connect over the network (localhost included).

**Fix:**

```
sudo setsebool -P httpd_can_network_connect on
# Perhaps a reload is not needed, but good practice wants us to test it anyways.
sudo systemctl restart rh-php72-php-fpm.service
sudo systemctl restart httpd.service
```

**RHEL/CentOS SELinux debug**

More often than not there might be issues with SELinux when not configured correctly. The below will give you pointers where to look and how to figure out what is wrong.

You can investigate SELinux issues without any tools by opening the audit log it generates. This log is found at /var/log/audit/audit.log. However, unless you know exactly what to look for and have a lot of free time, you’re going to find it difficult making sense of the log.

Install some handy tools:

```
# Note: This will pull in some X tools, you have been warned
sudo yum install setroubleshoot setools
```

We now have a tool called sealert that analyzes the audit log used by SELinux. Sealert will scan the log file and will then generate a report containing all discovered SELinux issues. In this overview of what went wrong you will see suggestions on how to fix them after the issue detected.

To run sealert from the command-line, we need to point it to the SELinux audit log.

```
sudo sealert -a /var/log/audit/audit.log
```

**Clearing the audit logs**

It is not recommended to clear the audit logs as they might contain information needed in the future for troubleshooting or security investigations. However, if that is not the case, just empty the audit log:

```
# > /var/log/audit/audit.log
```
**When to update MISP?**

One question might be how often to update MISP. You can update MISP as often as you like. If you see the following:

**MISP version**

Every version of MISP includes a json file with the current version. This is checked against the latest tag on github, if there is a version mismatch the tool will warn you about it. Make sure that you update MISP regularly.

<table>
<thead>
<tr>
<th>Currently installed version…</th>
<th>v2.4.97 (4462a72206a9cc5959c1facee90efdec2a308d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latest available version…</td>
<td>v2.4.97 (6e9b6fb80382346f338aa94f37b52d326b7cc551)</td>
</tr>
<tr>
<td>Status…</td>
<td>OK</td>
</tr>
<tr>
<td>Current branch…</td>
<td>2.4</td>
</tr>
</tbody>
</table>

[Update MISP]
This means that the main repository has an update available.

If you want to play it safer or want to integrate it in your Weekly/Bi-Monthly update routine you can track our Changelog a more up to date version is available here.

Further on we do regular tagged releases. (Approximately once per month.) The releases happen either if a milestone has been hit for a certain feature/improvement/fix or for any security related matters.

Thus you have the choice of either tracking 2.4 which is on a rolling release schedule, or track the tagged releases.

How to switch from tagged releases and back?

This can be achieved with the following git commands:

```
$ cd /var/www/MISP # aka. $PATH_TO_MISP
$ sudo -H -u www-data git checkout tags/$(git describe --tags 'git rev-list --tags --max-count=1')
```

## OS Upgrades

In theory all should "just work"(tm), but in practice the following dependencies might make your install unstable and need a little though before just doing the updates.

- `php/pear`
- `python`
- `apache`
- `init/scheme/scripts`
- `mariadb/mysql`
- `redis`
- `git`

### PHP

This is probably the most likely one that might get you into trouble.

The following happened on a Debian Testing lately. During the upgrade php got upgraded to php-7.3 and seemingly some php-7.2 dependencies were deinstalled and the system now had 2 concurrent versions of php installed.

The fix was to remove any "libapache2-mod-php7.2" packages and make sure that "apt remove libapache2-mod-php7.3" was installed. Most certainly you need to add symbolic links to "/etc/apache2/mods-enabled" to make php7.3 work.

Then double check if all the php dependencies are install, refer to the install documents.

The same for pear, where we mostly use 2 (bundled) packages: Console Command Line, Crypt GPG.

If you upgrade from a very old and out of date version of MISP this might raise issues.

    php.ini might also become problematic if you just erase the recommended defaults.

### Python3

If you use python2 for MISP, please read the install docs about MISP being Python 3 only.

Currently Python3.6 is minimum. It is known working on 3.7 with some minor difficulties (see PyMISP issues).

The biggest issue is certainly with PyMISP doig unexpected things when python might be updated.

Using a virtualenv, whilst not always ideal for all setups, will at least make sure that problems are contained a little more.

### Apache

Mostly config issues might be a show stopper. And major version updates where some underlying config might need to be changed.

### init/systemd

MISP launches a couple of things on boot. Changing what handles boot behaviour might have an impact.

### MariaDB/MySQL/redis

Similar to apache, most importantly always take good care that the DB engine is not all of a sudden changed wit
hout you noticing it. From minor to major updates, rarely things might need to be adapted.

```markdown
### git
Currently (as of v2.4.108) the git-cli command is used in MISP core. In very rare cases where the expected output changes, this might be an issue. Included here more as an FYI then anything else.
```

### Hardening

#### How do I harden my MISP instance?

You can check the [hardening section](https://misp.github.io/MISP/generic/hardening/) in the install guide.

#### Is there a MISP maintenance mode?

Yes, you want to flip your instances "Live-mode". This wants to be done on the CLI if you experience issues:

```bash
$PATH_TO_MISP/app/Console/cake "MISP.live" 0
```

Other related MISP Settings

Optional MISP.maintenance_message Great things are happening! MISP is undergoing maintenance, but will return shortly. You can contact the administration at $email or call CIRCL. The message that users will see if the instance is not live.

Critical MISP.live true Unless set to true, the instance will only be accessible by site-admins.

**Update MISP fails**

If your MISP instance is outdated, meaning ONLY the core, not the modules or dashboard or python modules, you well see the following.

**MISP version**

Every version of MISP includes a json file with the current version. This is checked against the latest tag on github, if there is a version mismatch the tool will warn you about it. Make sure that you update MISP regularly.

<table>
<thead>
<tr>
<th>Currently installed version</th>
<th>v2.4.96 (f3850747da103ca616a7dbaab955df373db272f7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latest available version</td>
<td>v2.4.97 (ce3c78cd7db60812d0147ced992a7650509d31da)</td>
</tr>
<tr>
<td>Status</td>
<td>Outdated version</td>
</tr>
<tr>
<td>Current branch</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td><strong>Update MISP</strong></td>
</tr>
</tbody>
</table>
Once you click on update MISP you will be asked confirmation.

Update MISP

Do you want to pull the latest commit from the branch? If you have made local changes to MISP the merge will fail.

Yes  No
If you are not on a branch, the UI will tell you this, the update will fail.

**Currently installed version.**  v2.4.96 (bbc8a0bf4dce4d64fb676a3a76bb4c6be091e890)

**Latest available version.**  v2.4.97 (ce3c78cd7db60812d0147ced992a7650509d31da)

**Status.** Outdated version

**Current branch.** You are not on a branch, Update MISP will fail

```
    git checkout app/composer.json 2>&1

    error: pathspec 'app/composer.json' did not match any file(s) known to git

    git pull origin 2>&1

    error: pathspec 'app/composer.json' did not match any file(s) known to git
    You are not currently on a branch.
    Please specify which branch you want to merge with.
    See git-pull(1) for details.

    git pull <remote> <branch>

    ==============

    git submodule update --init --recursive 2>&1
```
If you cannot write the `.git` files and directory as the user running the web server (and thus PHP), the update will fail. The following diagnostic check will let you know if you can update or not.

**Writeable Files**

/var/www/MISP-priv/app/Config/config.php…OK
/var/www/MISP-priv/.git/ORIG_HEAD…File is not writeable
In case you get a file not found on `.git/ORIG_HEAD`, this means that you have never updated your MISP OR you have installed git from an archive file (like .zip/.tar.gz or similar) Try to click update MISP and see what happens.

**Writable Files**

```
/var/www/MISP/app/Config/config.php…OK
/var/www/MISP/.git/ORIG_HEAD….File not found
```
What can go wrong if I update MISP?

In theory nothing. We put great effort into protecting the integrity of the data stored in your MISP instance. DB upgrades happen upon login or on reload once you have update the repository. You cannot “break” anything by clicking Update MISP worse case it will complain about something and you will certainly find the answer on this page.

If not, please open an issue on GitHub or come to our gitter chat to see if the community can help.

**error: pathspec 'app/composer.json' did not match any file(s) known to git**

This is **not** an error and can be ignore. Nothing will be impacted by this.

```
Currently installed version  v2.4.96 (f3850747da103ca616a7dbaab955df373db272f7)
Latest available version  v2.4.97 (ce3c78cd7db60812d0147ced992a7650509d31da)
Status  Outdated version
Current branch  2.4

```

```
git checkout app/composer.json 2>&1

error: pathspec 'app/composer.json' did not match any file(s) known to git

```

```
git pull origin 2.4 2>&1

error: pathspec 'app/composer.json' did not match any file(s) known to git

```

From https://github.com/MISP/MISP

* branch  2.4    -> FETCH_HEAD
  f3850747d..ce3c78cd7  2.4    -> origin/2.4

Updating f3850747d..ce3c78cd7

Fast-forward

  VERSION.json      | 2 +-
  app/Controller/AppController.php | 2 +-
  app/Controller/Component/ACLComponent.php | 1 +
  app/Controller/Component/RestResponseComponent.php | 16 ++-
  app/Controller/EventsController.php | 2 +-  
  app/Controller/SightingsController.php | 36 ++++++
  app/Lib/Export/CsvExport.php | 28 +++++
  app/Lib/Export/JsonExport.php | 9 +-
**MISP modules "Connection refused"**

**Module System**

This tool tests the various module systems and whether they are reachable based on the module settings.

<table>
<thead>
<tr>
<th>Module System</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enrichment</td>
<td>Connection refused</td>
</tr>
<tr>
<td>Import</td>
<td>Connection refused</td>
</tr>
<tr>
<td>Export</td>
<td>Connection refused</td>
</tr>
<tr>
<td>Cortex</td>
<td>System not enabled</td>
</tr>
</tbody>
</table>
If you get have a **Connection refused state** on your modules one of the following might be true.

- You have no **misp-modules** not installed
- They are instaled but not running
- Something completly different

If they are not installed, check out this section of the **INSTALL guide** of **misp-modules**.

In case they are not running, try this on the console:

```
sudo -u www-data /var/www/MISP/venv/bin/misp-modules -l 127.0.0.1 -s &
```

OR if you were foolish enough to not install in a Python virtualenv:

```
sudo -u www-data misp-modules -l 127.0.0.1 -s &
```

[warning] Running misp-modules like this will certainly kill it once you quit the session. Make sure it is in your `/etc/rc.local` or some ther init script that gets run on boot.

### Uninstalling MISP

There is no official procedure to uninstalling a MISP instance.

If you want to re-use a machine where MISP was installed, wipe the machine and do a fresh install. Consider the data in your MISP instance as potentially confidential and if you synchronized with other instances, be respectful and wipe it clean.

### Updating PyMISP to incorporate newer versions of the MISP object templates

In some cases, for instance if a newer version of a MISP object is present on the server but not yet on PyMISP, you want to reflect the current state in your PyMISP installation.

In order to do so, perform the following steps. It fetches the latest object templates and installs PyMISP again:

```
git clone https://github.com/MISP/PyMISP.git
cd PyMISP/pymisp/data
git submodule update --init
cd misp-objects
git pull origin master
cd .../...
sudo pip3 install -I .
```

### How to disable freetext/custom/user-created tags and only allow certain tags

Remove the "tag editor" from the permissions that you grant to users. Set all tags that you do not want to "hidden". There is a server setting to treat all incoming tags as hidden by default: `MISP.incoming_tags_disabled_by_default`

**Important** Make sure that you don't remove "tag editor" from sync users, or you'll be stripping tags from synchronized data.

### How to enable the csv import module?
First you have to enable the import services: double-click on "false" in the very first line and change it to "true".

In Server Settings & Maintenance -> Plugin Settings -> Import -> set "Plugin.Import_csvimport_enabled" to true. Afterwards you'll find the csvimport from within the newly created event: "Populate from..."

Don't use from the main site ("Import from...").

Why do I see 'The request has been black-holed' when I submit forms?

That's a security measure for form tampering protection.

All forms have a timeout (~15min) and all of them can only be submitted once. If you use your browser's "back" button and resubmit the form MISP will consider it as a potential attempt at form tampering.

Importing large feeds creates PHP Fatal error

When importing a large feed like the CIRCL feed, the job reaches 99% and then fails. The log file records:

```
PHP Fatal error: Allowed memory size of 536870912 bytes exhausted (tried to allocate 1941504 bytes) in /var/www/MISP/app/Model/Feed.php on line 691
```

In this case you will need to increase the memory_limit option in `php.ini` file

I deleted the admin user by mistake

Now, I only have Org Admin.

You have several options:

1. Delete the org admin. MISP automatically creates a new default site-admin user if no users are found in the db (mysql: truncate users;)

2. Upgrade a user to a site-admin, such as an org admin user:

   ```
   SELECT id, email from users;
   ```

   Note down the ID you want to upgrade. Let's say this is 2 for the example's sake.

   ```
   SELECT id, name from roles;
   ```

   Note down the role ID you want to upgrade. Let's say this is 1 for the example's sake.

   ```
   UPDATE users set role_id = 1 where id = 2;
   ```

config.php is not writeable

```
Warning: app/Config/config.php is not writeable. This means that any setting changes made here will NOT be saved.
```
According to the install guide, make sure to:

```bash
chown -R apache:apache /var/www/MISP
find /var/www/MISP -type d -exec chmod g=rx \;
chmod -R g+r,o= /var/www/MISP
```

If it still doesn’t work, make sure SELinux is not enabled or modify the rule set:

```bash
chcon -t httpd_sys_rw_content_t /var/www/MISP/app/files
chcon -t httpd_sys_rw_content_t /var/www/MISP/app/files/terms
chcon -t httpd_sys_rw_content_t /var/www/MISP/app/files/scripts/tmp
chcon -t httpd_sys_rw_content_t /var/www/MISP/app/Plugin/CakeResque/tmp
chcon -R -t httpd_sys_rw_content_t /var/www/MISP/app/tmp
chcon -R -t httpd_sys_rw_content_t /var/www/MISP/app/webroot/img/orgs
chcon -R -t httpd_sys_rw_content_t /var/www/MISP/app/webroot/img/custom
```

### How to debug misp-dashboard

This is the full chain from MISP to the live dashboard and some tips to find out which link is faulty.

1. **MISP** Ensure that ZMQ is installed and enabled with the correct settings
2. **MISP ZMQ** You can use MISP/tools/misp-zmq/sub.py which will subscribe to the ZMQ and print the data
3. **ZMQ subscriber** You can change the logging level from logging.INFO to logging.DEBUG and look in the logs for the string Pushed:`*` Or add a print statement in the put_in_redis_list function
4. **ZMQ dispatcher** Look in the logs for the string Handling:`*`
5. **Server (Flask) and Browser (live Dashboard)** Open the Web developer Network tab in your browser and look for the url `/_logs` with Content-Type: text/event-stream;

### How to update object templates?

```
git submodule update
```
in your MISP directory (or via the diagnostic page) and just click “Update Objects” in List Object Templates.

### What to do if my REST client is throwing SSL errors when trying to query my MISP instance?

The REST client will use the framework’s certificate store to validate the contacted host. If your root CA / self-signed certificate is not known by the certificate store, the request will fail. You can skip the SSL validation altogether using the “Skip SSL validation” checkbox.

### What to do if my REST client cannot reach the host, despite me being able to issue requests using Curl / Postman / etc.?

The REST client issues instructions to your MISP server to contact a remote host (most commonly itself). Always consider how your MISP server can address itself when using the REST client, by default it will prepend the requested relative path in the URL field with the instance’s baseurl.
If your MISP cannot reach itself via the baseurl the request will fail. You can use the "Use full path - disclose my API key" checkbox along with the full URL in the URL field to instruct MISP to use another path than what it would construct using the baseurl.

**How would one set up a sharing group with a remote org, where we only share a mutual community instance (i.e. we both have sync users on that instance). On our local instance, they exist as a remote org (from events that have synced from their instance via our shared community instance).**

It is not possible to do that. Keep in mind that if you are both on a mutual community instance, someone is in charge of that instance that will have database and admin level access. They would be able to inspect the data you exchange on their community instance with one another, so MISP will block any attempt to share with them.

If you really want to go through the community instance to exchange with them, you explicitly have to include the host organisation of the community instance (they would get access if they wanted to anyway, this way we can ensure that you are clear about that):

- You are org a on instance A.
- Your partner that you want to share with is org b on instance B.
- You have no way of directly reaching org b, but you both have access to instance C, which is run by org c (the sharing instance)
- In order to reach org b, you have two options for sharing groups, depending on whether you want to be able to push to them or want to rely on them pulling data from the community instance:

  **SG Option 1 (push all the way to B)**
  orgs: a, b, c
  instances, A, B, C

  **SG Option 2 (b has to pull from C):**
  orgs: a, b, c
  instances: A, C

**Is it possible to propose objects to an event?**

This is not possible yet. What you can do at the moment: Create a new event and extend it with the other (foreign) event.

**How to use the enforceWarninglist parameter in REST search?**

If you would like to export IoCs, for example into a suricata rule and exclude all values matching your warning lists, you can use the following:

```json
{
    "returnFormat": "suricata",
    "published": 0,
    "enforceWarninglist": 1
}
```

Keep in mind that unpublished events need the "published": 0 parameter in order to be exported.
Column not found issue

When a user attempts to add an object to an event and the following error is received (Level 1 debug enabled):

```sql
SQLSTATE[42S22]: Column not found: 1054 Unknown column 'Event.org_id' in 'where clause'
```

One potential resolution is to upgrade MISP to 2.4.107.

Symptoms

Users with the site-admin role are able to add objects to events without any error. This error was encountered when a user belonged to every role except site-admin.

WatchList Customization

**How to create a customized WatchList.**

WatchLists are stored within folder under `/var/www/MISP/app/files/warninglists/lists` Every folder contains a list.json file. Create a new folder and copy and modify an existing list (or create a new one from scratch). Ensure the "name" value within the file is unique. Increment the version number when the file is changed.

Within the MISP GUI, go to WarningLists and "Update WarningLists".

The new WarningList will now show up. In case of errors, check the permissions on the list.json and it's folder.

To modify the list or to add entries to it, go back to the file via the CLI, modify the file and reload it via the GUI ("Update WarningLists").

How to upgrade PHP on RHEL/CentOS?

To our knowledge, there is no way to "upgrade" PHP. You'll need to install the new PHP version like you're doing a fresh install. You may try copying your old `php.ini` to your new PHP config directory which may work. We would recommend redoing the config though.

**Example: Upgrade from PHP 7.2 to 7.3 on CentOS 7**

Enable repository

```
$ sudo yum install -y http://rpms.remirepo.net/enterprise/remi-release-7.rpm
$ sudo yum-config-manager --enable remi-php73
```

Install packages

```
```

Confirm GPG key if required:

```bash
Retrieving key from file://etc/pki/rpm-gpg/RPM-GPG-KEY-remi
Importing GPG key 0x00F97F56:
  Userid : "Remi Collet <RPMS@FamilleCollet.com>"
  Fingerprint: 1ee0 4ce4 8844 a44a 259a 5df5 084e 6f47 0ff9 ff56
```
Install required PEAR-modules

```
$ sudo php73-pear channel-update pear.php.net
Updating channel "pear.php.net"
Update of Channel "pear.php.net" succeeded
$ sudo php73-pear install /var/www/MISP/INSTALL/dependencies/Console_CommandLine/package.xml
install ok: channel://pear.php.net/Console_CommandLine-1.2.2
$ sudo php73-pear install /var/www/MISP/INSTALL/dependencies/Crypt_GPG/package.xml
install ok: channel://pear.php.net/Crypt_GPG-1.6.3
```

PHP configuration

```
Edit /etc/opt/remi/php73/php.ini:

date.timezone = "Europe/Berlin"
max_execution_time = 300
memory_limit = 512M
upload_max_filesize = 50M
post_max_size = 50M
```

Switch to PHP 7.3

```
$ sudo systemctl stop rh-php70-php-fpm
$ sudo systemctl start php73-php-fpm
# check if everything's fine
$ sudo systemctl status php73-php-fpm
```

Now check if the MISP web UI is accessible and if the diagnostics page shows any errors.

The diagnostics page will show "PHP CLI Version (>7.2 recommended): Unknown (Issues determining version)"). That's a known issue.

Disable/enable services

```
$ sudo systemctl disable rh-php70-php-fpm
$ sudo systemctl enable php73-php-fpm
```

How to add a galaxy to an event via PyMISP

A galaxy can be assigned like a tag. You can use the add tag function and copy the full connector-tag. Example `misp-galaxy:ransomware="Locky"`, which can be found in `/galaxy_clusters/view/`

Updating PHP from 7.2 to 7.4.5 on Ubuntu 18.04

Installation
1. Disable and Uninstall Currently Installed SSDEEP

   sudo phpdismod ssdeep
   sudo pecl uninstall ssdeep
   sudo apt purge ssdeep
   sudo rm -rf /etc/php/7.2/mods-available/ssdeep.ini

2. Install PHP 7.4.5

   sudo apt install software-properties-common -qy
   sudo add-apt-repository ppa:ondrej/php -y
   sudo apt update
   sudo apt install -qy
   libapache2-mod-php7.4
   php7.4-cli
   php7.4-dev
   php7.4-json
   php7.4-xml
   php7.4-mysql
   php7.4-opcache
   php7.4-readline
   php7.4-mbstring
   php-redis
   php-gmpg
   php-gd
   sudo apt update
   sudo apt upgrade -y

3. Install SSDEEP

   cd /usr/local/src
   sudo tar zxvf ssdeep-2.14.1.tar.gz
   cd ssdeep-2.14.1
   sudo ./configure --datadir=/usr --prefix=/usr --localstatedir=/var --sysconfdir=/etc
   sudo make
   sudo make install

4. Test SSDEEP

   ssdeep -h

5. Install ssdeep_php

   sudo pecl channel-update pecl.php.net
   sudo pecl install ssdeep

6. Enable SSDEEP in both 7.2 and 7.4 (as root sudo su)

   echo 'extension=ssdeep.so' > /etc/php/7.2/mods-available/ssdeep.ini
   echo 'extension=ssdeep.so' > /etc/php/7.4/mods-available/ssdeep.ini

7. Enable SSDEEP PHP Mod

   sudo phpenmod ssdeep
8. Set PHP 7.4.5 to default PHP

```bash
sudo a2dismod php7.2
sudo a2enmod php7.4
sudo update-alternatives --set php /usr/bin/php7.4
```

9. [Optional] Set better values for defaults

```bash
sudo sed -i "s/max_execution_time = 30/max_execution_time = 300/" /etc/php/7.4/apache2/php.ini ;
sudo sed -i "s/memory_limit = 128M/memory_limit = 2048M/" /etc/php/7.4/apache2/php.ini ;
sudo sed -i "s/upload_max_filesize = 2M/upload_max_filesize = 500M/" /etc/php/7.4/apache2/php.ini ;
sudo sed -i "s/post_max_size = 8M/post_max_size = 500M/" /etc/php/7.4/apache2/php.ini ;
sudo sed -i "s/max_execution_time = 30/max_execution_time = 300/" /etc/php/7.4/cli/php.ini ;
sudo sed -i "s/upload_max_filesize = 2M/upload_max_filesize = 500M/" /etc/php/7.4/cli/php.ini ;
sudo sed -i "s/post_max_size = 8M/post_max_size = 5000M/" /etc/php/7.4/cli/php.ini ;
```

10. Restart Apache to implement changes

```bash
sudo sudo systemctl restart apache2
```

**Verification of php 7.2 to 7.4**

1. Administration > Server Settings & Maintenance

2. Diagnostics

3. Scroll down to the PHP Settings section and verify

**What are the required steps after a MISP installation to have a properly running instance?**

- First login with the installation credentials and change the password immediately (especially if your instance is publicly accessible)
- Set the base_url to the hostname of your machine (apache virtualhost name)
- Create a new organisation which will be the host organisation running the MISP instance
- Set the new organisation in `MISP.host_org_id` to replace the default one
- Set messages like `MISP.footermidleft` and alike to a proper message to help your users
- Create a new user as admin role with the new organisation
- Log with the new user, if successful, remove the default user used during the installation such as `admin@admin.test`
- Select and enable required taxonomies for your sharing community
- Select and enable the external feeds (as caching only if you don't want full events but you can get the full feeds too)
- Select and enable the warning-list (if you don't know what to enable, select all)
- Add the remote MISP instances where you have access to (either caching only or full pull if you want the complete events)
Developer FAQ

Main Developer Resources

The main developer resources can be found on GitHub in the MISP Wiki.

The following pages are worth inspecting closer in case you want to actively develop for MISP:

- The real FAQ
- Contributor Overview
- Some objectives of MISP
- Various deployment tools
- MISP Code of Conduct
- UI coloring scheme
- Notes on MISP and STIX 2
- Commit Messages Best Practices
- Internationalization (i18n)

Our gitter channel is a welcome place to ask other community developers in case you are stuck.
Appendices

Summary

- Appendix A: External Authentication
  - [The external authentication mechanism described](#the-external-authentication-mechanism-described)
  - [Setting up the external authentication mechanism](#setting-up-the-external-authentication-mechanism)
  - [User management](#user-management)
  - [Logging](#logging)

- Appendix B: ACL descriptors
  - [Querying the ACL system](#querying-the-acl-system)
  - [Getting a list of URLs accessible to a role](#getting-a-list-of-urls-accessible-to-a-role)
  - [Getting a list of all accessible controllers and actions in MISP](#getting-a-list-of-all-accessible-controllers-and-actions-in-misp)
  - [Viewing a list of yet unmapped functions](#viewing-a-list-of-yet-unmapped-functions)

- Appendix C: Official MISP developments
- Appendix D: Third-party development
- Appendix E: Other Threat Intel Resources
- Appendix F: LDAP Authentication
  - [Installation and configuration](#installation-and-configuration)
  - [Debugging](#debugging)
  - [Migrating existing user to LDAP](#migrating-existing-user-to-ldap)
  - [Caveats](#caveats)

Appendix A: External Authentication

**The external authentication mechanism described**

The external authentication allows a user or an external tool to authenticate with MISP using an arbitrary value passed along in a custom header. This authentication method overrides the regular authentication mechanisms and is customisable by a site-admin.

It is possible to create a mixed mode MISP setup where certain users can go through the normal authentication mechanism and other users are required to use the external authentication method.

**Setting up the external authentication mechanism**

To change the authentication settings, navigate to Administration - Server settings - Plugin settings

The settings associated with the external authentication can be found by pressing the CustomAuth button as depicted below:
### CustomAuth

<table>
<thead>
<tr>
<th>Priority</th>
<th>Setting</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional</td>
<td>Plugin CustomAuth_enable</td>
<td>true</td>
<td>Enable this fun authenticate wi</td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin CustomAuth_header</td>
<td>radac_auth_header</td>
<td>Set the header header.</td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin CustomAuth_required</td>
<td>false</td>
<td>If this setting is</td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin CustomAuth_only_allow_source</td>
<td></td>
<td>If you are using</td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin CustomAuth_name</td>
<td>Radac</td>
<td>The name of the creation page.</td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin CustomAuth_disable_log</td>
<td>true</td>
<td>Disable the log</td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin CustomAuth_password_reset</td>
<td><a href="https://my/custom/password_reset">https://my/custom/password_reset</a></td>
<td>Provide your custom_password_reset</td>
</tr>
<tr>
<td>Optional</td>
<td>Plugin CustomAuth_logout</td>
<td><a href="https://my/custom/logout">https://my/custom/logout</a></td>
<td>Provide a custs system you use</td>
</tr>
</tbody>
</table>
To change a setting simply double click on the value to edit the field. Use the guidance provided by the setting tool to configure the external authentication. The accessible settings are as follows:

- **enable**: Enable or disable external authentication (off by default)
- **header**: The header which MISP will use to identify users
- **required**: Enabling this setting will force all users to use the external authentication. Leave this disabled allows administrators to assign external authentication or regular authentication users.
- **only_allow_source**: Setting a url / IP address here will only allow requests that originated from the given address
- **name**: The name to be used for the authentication mechanism. This is reflected in the user creation / edit views, the logs and the error messages on failed logins.
- **disable_logout**: Disable the default logout button. Using an external authentication mechanism that authenticates via the header with each requests makes the logout button obsolete.
- **custom_password_reset**: If your authentication system has a url that a user can access to reset his/her password, please specify the full url for it here. This will then be reused in the UI.
- **custom_password_logout**: If your authentication system has a url that a user can access to logout, please specify the full url for it here. This will then be reused in the UI.

**User management**

Using a new setting, user self management can be disabled for all users that are not administrators via the MISP.disableUserSelfManagement setting, found in the MISP settings tab. Enabling this setting removes the ability of users to change their user settings and reset their authentication keys. All other functionality remains unchanged.
Email

☐ External authentication user
☐ Set password

Organisation

Choose organisation

Role

Site Admin

Authkey

DdeSSGRSN9vSS9pbGEupf0d9ic

Sync user for

Not bound to a server

GPG key

Fetch GPG key

☐ Receive alerts when events are published

☐ Receive alerts from "contact reporter" requests

☐ Disable this user account

Submit
To create an external authenticated user, simply tick the External authentication user checkbox, after which an external auth key field will appear. This will be used to identify the users via the passed along header.

**Logging**

For a description of the logging facilities provided by this plugin, please refer to the "Logging of failed authentication attempts" section of the Administration section.

**Appendix B: ACL descriptors**

**Querying the ACL system**

MISP allows site admins to query the ACL system for various types of data. This can be interesting when tuning for example WAF access to MISP. All applicable queries can be requested via /servers/queryACL

**Getting a list of URLs accessible to a role**

https://<misp url>/servers/queryACL/printRoleAccess/<role id>

The above URL will return a JSON with all accessible URLs for the given role ID. If no Role ID is provided, a JSON containing all roles and their access lists will be returned.

**Example:**

```json
{
  "2": {
    "name": "User",
    "urls": [
      "/attributes/add/",
      "/attributes/add_attachment/",
      "/attributes/add_threatconnect/",
      "/attributes/attributeReplace/",
      "/attributes/delete/",
      "/attributes/deleteSelected/",
      "/attributes/download/",
      "/attributes/downloadAttachment/",
      "/attributes/downloadSample/",
      "/attributes/edit/",
      "/attributes/editField/",
      "/attributes/editSelected/",
      "/attributes/fetchEditForm/",
      "/attributes/fetchViewValue/",
      "/attributes/hoverEnrichment/",
      "/attributes/index/",
      "/attributes/restSearch/",
      "/attributes/returnAttributes/",
      "/attributes/rpz/",
      "/attributes/search/",
      "/attributes/searchAlternate/",
      "/attributes/text/",
      "/attributes/updateAttributeValues/",
      "/attributes/view/",
      "/eventDelegations/acceptDelegation/",
      "/eventDelegations/delegateEvent/",
      "/eventDelegations/deleteDelegation/",
      "/eventDelegations/view/",
      "/events/add/",
      "/events/addIOC/",
      "/events/addTag/"
    ]
  }
}
Getting a list of all accessible controllers and actions in MISP

https://<misp url>/servers/queryACL/printAllFunctionNames

This URL will return a JSON with all controller and all mapped functions within them.

Viewing a list of yet unmapped functions

https://<misp url>/servers/queryACL/findMissingFunctionNames

Functions that have not been tied into the new ACL yet show up here. These functions will (until added to the ACL) only be accessible to site admins.

Appendix C: Official MISP developments

This section lists the projects that can be found on the main MISP GitHub page e know of but not officially support and rely on their respective maintainers to keep up to date to the MISP 2.4 developments.

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>misp-objects</td>
<td>Definition, description and relationship types of MISP objects</td>
<td>Core to MISP, frequently updated and tested</td>
</tr>
</tbody>
</table>
Appendix D: Third-party development

This section lists some projects we know of but not officially support and rely on their respective maintainers to keep up to date to the MISP 2.4 developments.

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>MISP-STIX-ESM</td>
<td>Exports MISP events to STIX and ingest into McAfee ESM</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>Docker MISP</td>
<td>Automated Docker MISP container</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>misp42splunk</td>
<td>A Splunk app to use MISP in background and combine with TheHive</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>getmispnoc</td>
<td>getiocmisp is a Splunk custom search command that helps to extract IOCs from a MISP instance.</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>OTX MISP</td>
<td>Imports Alienvault OTX pulses to a MISP instance</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>BTG</td>
<td>BTG’s purpose is to make fast and efficient search on IOC</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>MISP OSINT Collection</td>
<td>Collection of best practices to add OSINT into MISP and/or MISP communities</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>IBM XFE module</td>
<td>Various IBM X-Force Exchange modules</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>MISP dockerized</td>
<td>MISP dockerized is a project designed to provide an easy-to-use and easy-to-install 'out of the box' MISP instance that includes everything you need to run MISP with minimal host-side requirements.</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>MISP dockerized modules</td>
<td>MISP-modules for MISP dockerized</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>FireMISP</td>
<td>FireEye Alert json files to MISP Malware information sharing platfrom (Alpha)</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>MISP Chrome Plugin</td>
<td>MISP Chrome plugin for adding and looking up indicators</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>PySight2MISP</td>
<td>PySight2MISP is a project that can be run to be used as glue between iSight intel API and MISP API</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>tie2misp</td>
<td>Import DCSO TIE IOCs as MISP events</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>security onion MISP</td>
<td>Grab NIDS rules and Bro Intel generated from a MISP instance and use them in Security Onion</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>virustream</td>
<td>A script to track malware IOCs with OSINT on Twitter.</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>LAC CSV Import</td>
<td>Register MISP events based on information described in files such as CSV and TSV.</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>The Hive</strong></td>
<td><strong>TheHive</strong>: a Scalable, Open Source and Free Security incident Response Platform</td>
<td>tested and known working</td>
</tr>
<tr>
<td><strong>puppet-misp</strong></td>
<td>This module installs and configures MISP - puppet forge site</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>Ansible MISP</strong></td>
<td>Ansible playbook to install Malware Information Sharing Platform (MISP)</td>
<td>unmaintained</td>
</tr>
<tr>
<td><strong>ansible MISP</strong></td>
<td>ansible role to setup MISP</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>OpenDXL ATD MISP</strong></td>
<td>Automated threat intelligence collection with McAfee ATD, OpenDXL and MISP</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>IMAP Proxy</strong></td>
<td>Modular IMAP proxy (including PyCIRCLeanMail and MISP forward modules)</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>AutoMISP</strong></td>
<td>automate your MISP installs - This shell script is designed to automatically install MISP and the misp-modules extension on either Ubuntu 16.04, or 18.04.</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>Palo Alto Networks report_to_misp</strong></td>
<td>Parse a report and import the events into MISP</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>Palo Alto Networks minemeld-misp</strong></td>
<td>MineMeld nodes for MISP</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>golang-misp</strong></td>
<td>Golang Library to interact with your MISP instance</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>go-misp</strong></td>
<td>Golang MISP API Client</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>MISP MAR</strong></td>
<td>Integration between MISP platform and McAfee Active Response</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>MISP IOC Validator</strong></td>
<td>Validate IOC from MISP ; Export results and iocs to SIEM and sensors using syslog and CEF format</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>vt2misp</strong></td>
<td>Script to fetch data from virustotal and add it to a specific event as an object</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>Threat Pinch Lookup</strong></td>
<td>Documentation and Sharing Repository for ThreatPinch Lookup Chrome &amp; Firefox Extension</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>dovehawk</strong></td>
<td>Dovehawk is a Bro module that automatically imports MISP indicators and reports Sightings</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>yara-exporter</strong></td>
<td>Exporting MISP event attributes to yara rules usable with Thor apt scanner</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>volatility-misp</strong></td>
<td>Volatility plugin to interface with MISP</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>misp2bro</strong></td>
<td>Python script that gets IOC from MISP and converts it into BRO intel files.</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>TA-misp</strong></td>
<td>Splunk integration with MISP</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>MISP QRadar</strong></td>
<td>The Project can used to integrate QRadar with MISP Threat Sharing Platform</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>pymisp-suricata_search</strong></td>
<td>Multi-threaded suricata search module for MISP</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td><strong>MISP-</strong></td>
<td>Script to interface MISP with Facebook ThreatExchange</td>
<td>Not tested by MISP</td>
</tr>
<tr>
<td>ThreatExchange</td>
<td>Script to interface MISP with Facebook ThreatExchange</td>
<td>core team</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>apty</td>
<td>Automated Payload Test Controller</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>aptmap</td>
<td>A map displaying threat actors from the misp-galaxy</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>mispy</td>
<td>Another MISP module for Python</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>MispSharp</td>
<td>C# Library for MISP</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>mispbtc</td>
<td>get BTC addresses from MISP and fetch BTC transactions</td>
<td>Tested by MISP core team</td>
</tr>
<tr>
<td>Privacy Aware</td>
<td>Master Thesis including MISP data.</td>
<td>Master thesis</td>
</tr>
<tr>
<td>Sharing of IoCs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>in MISP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sam-bot</td>
<td>Bot to create MISP events from data in Slack</td>
<td>Not tested by MISP core team</td>
</tr>
<tr>
<td>Polarity.io</td>
<td>&quot;Polarity is the memory augmentation platform that makes your team smarter”</td>
<td>Not tested by MISP core team</td>
</tr>
</tbody>
</table>

**Appendix E: Other Threat Intel Ressources**

A brief list of online resources that around #ThreatIntel

- A curated list of awesome malware analysis tools and resources. Inspired by awesome-python and awesome-php.
- An authoritative list of awesome devsecops tools with the help from community experiments and contributions.DEV.SEC.OPS
- Advance Python IOC extractor

**Appendix F: LDAP Authentication**

MISP supports LDAP authentication from version 2.4.xxx. This manual will show how to configure LDAP authentication.

**Installation and configuration**

1. Install `mod_ldap` PHP module

   ```bash
   # for Centos or RHEL
   yum install rh-php72-php-ldap
   # for Ubuntu or debian
   apt install php-ldap
   ```

2. Prepare variables for configuration

3. `- a full LDAP URI of server. For example: ldap://example.com`.

4. `- DN for path that contains users. For example: cn=users,cn=accounts,dc=example,dc=com`.

5. `- user that can read. For example: uid=misp,cn=sysaccounts,cn=etc,dc=example,dc=com`.

6. `- password for that user.`

7. `- group with access to MISP. For example: cn=misp-users,cn=groups,cn=accounts,dc=example,dc=com`.
8. Configure MISP ApacheSecureAuth in app/Config/config.php

```
'LdapAuth' => array,
    'enabled' => true,
    'name' => 'My Identity provider',
    'ldapServer' => '',
    'ldapDN' => '',
    'ldapSearchFilter' => '(objectclass=inetuser)',
    'ldapReaderUser' => '',
    'ldapReaderPassword' => '',
    'ldapUserGroup' => '',
    'updateUser' => true,
);
```

Required variables:

- **enabled** – if it is true, all users must log in through LDAP account.
- **ldapServer** – a full LDAP URI of the form ldap://hostname:port or ldaps://hostname:port for TLS encryption.
- **ldapDN** – DN for a path that contains users.

Optional variables:

- **name** – identity provider name. Will be shown in the login screen and user editing for. Can contain HTML.
- **ldapReaderUser** – DN or RDN LDAP user with permission to read LDAP information about users.
- **ldapReaderPassword** – password for that user.
- **ldapSearchFilter** – LDAP search filter.
- **ldapSearchAttribute** – LDAP attribute that contains username. Default: **uid**.
- **ldapEmailField** – LDAP attribute (string) or attributes (array) that will be checked if contains user e-mail address. If you want to change or add field, you should also add that field/fields to **ldapAttributes**. Default: **mail**.
- **ldapAttributes** – fields that will be fetched from LDAP server. Default: **mail** and **memberof**.
- **ldapUserGroup** – LDAP group that must be assigned to user to access MISP. Default: not set.
- **createUser** - if true, MISP will create new user from LDAP. Default true.
- **updateUser** - if true, MISP will update existing users information (e-mail address and role) from LDAP after login. Default: false.
- **ldapDefaultOrg** – default organization ID for user from LDAP. By default it is the first organization in the database.
- **ldapDefaultRoleId** - default role for newly created user. It can be integer or array when key contains LDAP group and value assigned role ID. Must be defined if **updateUser** is set to true (without that variable, user will be disabled).
- **ldapNetworkTimeout** - timeout for communication with LDAP server in seconds. Default: 5 seconds.
- **ldapIgnoreReferrals** - follow referrals returned by the LDAP server. Default: false.
- **ldapStartTls** - enable STARTTLS. Default: true.

**Debugging**

Setting LDAP authentication can be sometimes tricky. For debugging, you can check MISP error log (by default in /var/www/MISP/app/tmp/logs/error.log) or debug log (by default in /var/www/MISP/app/tmp/logs/debug.log) that can contain useful information with problem description.

**Migrating existing user to LDAP**

Because LDAP and MISP users are paired by e-mail address, it is possible to migrate existing user account to LDAP managed. When you enable LDAP support and LDAP user will try to log in, an existing user in MISP with the same e-mail address will be found and then assigned to LDAP user.

**Caveats**
● When a user is disabled in LDAP or is removed from the required group, it will be not automatically disabled in MISP. That means that user will be disabled when he tries to login (with form or with Auth key), but for example, notification e-mails will still work until he tries to log in.

● When a user is disabled in LDAP and also in MISP and then enabled in LDAP, it will be enabled in MISP for next login just when `updateUser` is set to `true`.

● Currently it is not possible to log in with both LDAP and local (MISP) accounts.

● Admins can change users email address. But when `updateUser` is set to true, when the user will log in again, the e-mail address will be updated from LDAP.

● `Security.require_password_confirmation` setting currently doesn't work with LDAP authentication. But on the other hand, since user cannot change e-mail address and password, this setting is not important.