Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 18, 2017

A. Dulaunoy
A. Iklody
CIRCL

October 15, 2016

# MISP core format
# draft-dulaunoy-misp-core-format-00

## Abstract

This document describes the MISP core format used to exchange indicators and threat information between MISP (Malware Information and threat Sharing Platform) instances. The JSON format includes the overall structure along with the semantic associated for each respective key. The format is described to support other implementations which reuse the format and ensuring an interoperability with existing MISP [MISP-P] software and other Threat Intelligence Platforms.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 18, 2017.

## Copyright Notice

Table of Contents



## 1. Introduction

Sharing threat information became a fundamental requirements in the Internet, security and intelligence community at large. Threat information can include indicators of compromise, malicious file indicators, financial fraud indicators or even detailed information about a threat actor. MISP [MISP-P] started as an open source project in late 2011 and the MISP format started to be widely used as an exchange format within the community in the past years. The aim of this document is to describe the specification and the MISP core format.

### 1.1. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## 2. Format

### 2.1. Overview

The MISP core format is in the JSON [RFC4627] format. In MISP, an event is composed of a single JSON object.

A capitalized key (like Event, Org) represent a data model and a non-capitalized key is just an attribute. This nomenclature can support an implementation to represent the MISP format in another data structure.

### 2.2. Event

An event is a simple meta structure scheme where attributes and meta-data are embedded to compose a coherent set of indicators. An event can be composed from an incident, a security analysis report or a specific threat actor analysis. The meaning of an event only depends of the information embedded in the event.

#### 2.2.1. Event Attributes

##### 2.2.1.1. uuid

uuid represents the Universally Unique IDentifier (UUID) [RFC4122] of the event. The uuid MUST be preserved for any updates or transfer of the same event. UUID version 4 is RECOMMENDED when assigning it to a new event.

uuid is represented as a JSON string. uuid MUST be present.

##### 2.2.1.2. id

id represents the human-readable identifier associated to the event for a specific MISP instance.

id is represented as a JSON string. id SHALL be present.

#### 2.2.1.3. published

published represents the event publication state. If the event was published, the published value MUST be true. In any other publication state, the published value MUST be false.

published is represented as a JSON boolean. published MUST be present.

#### 2.2.1.4. info

info represents the information field of the event. info a free-text value to provide a human-readable summary of the event. info SHOULD NOT be bigger than 256 characters and SHOULD NOT include new-lines.

info is represented as a JSON string. info MUST be present.

#### 2.2.1.5. threat_level_id

threat_level_id represents the threat level.

0:
   Undefined

1:
   Low

2:
   Medium

3:
   High

If a higher granularity is required, a MISP taxonomy applied as a Tag SHOULD be preferred.

threat_level_id is represented as a JSON string. threat_level_id SHALL be present.

#### 2.2.1.6. analysis

analysis represents the analysis level.

0:
   Initial

1:
   Ongoing

2:
Complete

If a higher granularity is required, a MISP taxonomy applied as a Tag SHOULD be preferred.

analysis is represented as a JSON string. analysis SHALL be present.

#### 2.2.1.7. date

date represents a reference date to the event in ISO 8601 format (date only: YYYY-MM-DD). This date corresponds to the date the event occured, which may be in the past.

date is represented as a JSON string. date MUST be present.

#### 2.2.1.8. timestamp

timestamp represents a reference time when the event, or one of the attributes within the event was created, or last updated/edited on the instance. timestamp is expressed in seconds (decimal) since 1st of January 1970 (Unix timestamp). The time zone MUST be UTC.

timestamp is represented as a JSON string. timestamp MUST be present.

#### 2.2.1.9. publish_timestamp

publish_timestamp represents a reference time when the event was published on the instance. published_timestamp is expressed in seconds (decimal) since 1st of January 1970 (Unix timestamp). At each publication of an event, publish_timestamp MUST be updated. The time zone MUST be UTC.

publish_timestamp is represented as a JSON string. publish_timestamp MUST be present.

#### 2.2.1.10. org_id

org_id represents a human-readable identifier referencing an Org object of the organization which generated the event.

The org_id MUST be updated when the event is generated by a new instance.

org_id is represented as a JSON string. org_id MUST be present.

#### 2.2.1.11. orgc_id

orgc_id represents a human-readable identifier referencing an Orgc object of the organization which created the event.

The orgc_id and Orc object MUST be preserved for any updates or transfer of the same event.

orgc_id is represented as a JSON string. orgc_id MUST be present.

#### 2.2.1.12. attribute_count

attribute_count represents the number of attributes in the event. attribute_count is expressed in decimal.

attribute_count is represented as a JSON string. attribute_count SHALL be present.

#### 2.2.1.13. distribution

distribution represents the basic distribution rules of the event. The system must adhere to the distribution setting for access control and for dissemination of the event.

distribution is represented by a JSON string. distribution MUST be present and be one of the following options:

0
: Your Organisation Only

1
: This Community Only

2
: Connected Communities

3
: All Communities

4
: Sharing Group

#### 2.2.1.14. sharing_group_id

sharing_group_id represents a human-readable identifier referencing a Sharing Group object that defines the distribution of the event, if distribution level "4" is set.

sharing_group_id is represented by a JSON string and MUST be present. If a distribution level other than "4" is chosen the sharing_group_id MUST be set to "0".

## 2.3. Objects

### 2.3.1. Org

An Org object is composed of an uuid, name and id.

The uuid represents the Universally Unique IDentifier (UUID) [RFC4122] of the organization. The organization UUID is globally assigned to an organization and SHALL be kept overtime.

The name is a readable description of the organization and SHOULD be present. The id is a human-readable identifier generated by the instance and used as reference in the event.

uuid, name and id are represented as a JSON string. uuid, name and id MUST be present.

#### 2.3.1.1. Sample Org Object

```
"Org": {
        "id": "2",
        "name": "CIRCL",
        "uuid": "55f6ea5e-2c60-40e5-964f-47a8950d210f"
       }
```

### 2.3.2. Orgc

An Orgc object is composed of an uuid, name and id.

The uuid MUST be preserved for any updates or transfer of the same event. UUID version 4 is RECOMMENDED when assigning it to a new event. The organization UUID is globally assigned to an organization and SHALL be kept overtime.

The name is a readable description of the organization and SHOULD be present. The id is a human-readable identifier generated by the instance and used as reference in the event.

uuid, name and id are represented as a JSON string. uuid, name and id MUST be present.

## 2.4. Attribute

Attributes are used to describe the indicators and contextual data of an event. The main information contained in an attribute is made up of a category-type-value triplet, where the category and type give meaning and context to the value. Through the various category-type combinations a wide range of information can be conveyed.

A MISP document MUST at least includes category-type-value triplet described in section "Attribute Attributes".

### 2.4.1. Sample Attribute Object

```
"Attribute": {
              "id": "346056",
              "type": "comment",
              "category": "Other",
              "to_ids": false,
              "uuid": "57f4f6d9-cd20-458b-84fd-109ec0a83869",
              "event_id": "3357",
              "distribution": "5",
              "timestamp": "1475679332",
              "comment": "",
              "sharing_group_id": "0",
              "deleted": false,
              "value": "Hello world",
              "SharingGroup": [],
              "ShadowAttribute": [],
              "RelatedAttribute": []
             }
```

### 2.4.2. Attribute Attributes

#### 2.4.2.1. uuid

uuid represents the Universally Unique IDentifier (UUID) [RFC4122] of the event. The uuid MUST be preserved for any updates or transfer of the same event. UUID version 4 is RECOMMENDED when assigning it to a new event.

uuid is represented as a JSON string. uuid MUST be present.

#### 2.4.2.2. id

id represents the human-readable identifier associated to the event for a specific MISP instance.

id is represented as a JSON string. id SHALL be present.

#### 2.4.2.3. type

type represents the means through which an attribute tries to describe the intent of the attribute creator, using a list of pre-defined attribute types.

type is represented as a JSON string. type MUST be present and it MUST be a valid selection for the chosen category. The list of valid category-type combinations is as follows:

Internal reference
text, link, comment, other

Targeting data
target-user, target-email, target-machine, target-org, target-location, target-external, comment

Antivirus detection
link, comment, text, attachment, other

Payload delivery
md5, sha1, sha224, sha256, sha384, sha512, sha512/224, sha512/256, ssdeep, imphash, authentihash, pehash, tlsh, filename, filename|md5, filename|sha1, filename|sha224, filename|sha256, filename|sha384, filename|sha512, filename|sha512/224, filename|sha512/256, filename|authentihash, filename|ssdeep, filename|tlsh, filename|imphash, filename|pehash, ip-src, ip-dst, hostname, domain, email-src, email-dst, email-subject, email-attachment, url, user-agent, AS, pattern-in-file, pattern-in-traffic, yara, attachment, malware-sample, link, malware-type, comment, text, vulnerability, x509-fingerprint-sha1, other

Artifacts dropped
md5, sha1, sha224, sha256, sha384, sha512, sha512/224, sha512/256, ssdeep, imphash, authentihash, filename, filename|md5, filename|sha1, filename|sha224, filename|sha256, filename|sha384, filename|sha512, filename|sha512/224, filename|sha512/256, filename|authentihash, filename|ssdeep, filename|tlsh, filename|imphash, filename|pehash, regkey, regkey|value, pattern-in-file, pattern-in-memory, pdb, yara, attachment, malware-sample, named pipe, mutex, windows-scheduled-task, windows-service-name, windows-service-displayname, comment, text, x509-fingerprint-sha1, other

Payload installation
md5, sha1, sha224, sha256, sha384, sha512, sha512/224, sha512/256, ssdeep, imphash, authentihash, pehash, tlsh, filename, filename|md5, filename|sha1, filename|sha224, filename|sha256,

filename|sha384, filename|sha512, filename|sha512/224, filename|sha512/256, filename|authentihash, filename|ssdeep, filename|tlsh, filename|imphash, filename|pehash, pattern-in-file, pattern-in-traffic, pattern-in-memory, yara, vulnerability, attachment, malware-sample, malware-type, comment, text, x509-fingerprint-sha1, other

Persistence mechanism
filename, regkey, regkey|value, comment, text, other

Network activity
ip-src, ip-dst, hostname, domain, domain|ip, email-dst, url, uri, user-agent, http-method, AS, snort, pattern-in-file, pattern-in-traffic, attachment, comment, text, x509-fingerprint-sha1, other

Payload type
comment, text, other

Attribution
threat-actor, campaign-name, campaign-id, whois-registrant-phone, whois-registrant-email, whois-registrant-name, whois-registrar, whois-creation-date, comment, text, x509-fingerprint-sha1, other

External analysis
md5, sha1, sha256, filename, filename|md5, filename|sha1, filename|sha256, ip-src, ip-dst, hostname, domain, domain|ip, url, user-agent, regkey, regkey|value, AS, snort, pattern-in-file, pattern-in-traffic, pattern-in-memory, vulnerability, attachment, malware-sample, link, comment, text, x509-fingerprint-sha1, other

Financial fraud
btc, iban, bic, bank-account-nr, aba-rtn, bin, cc-number, prtn, comment, text, other

Other
comment, text, other

Attributes are based on the usage within their different communities. Attributes can be extended on a regular basis and this reference document is updated accordingly.

#### 2.4.2.4. category

category represents the intent of what the attribute is describing as selected by the attribute creator, using a list of pre-defined attribute categories.

category is represented as a JSON string. category MUST be present and it MUST be a valid selection for the chosen type. The list of valid category-type combinations is mentioned above.

#### 2.4.2.5. to_ids

to_ids represents whether the attribute is meant to be actionable. Actionable defined attributes that can be used in automated processes as a pattern for detection in Local or Network Intrusion Detection System, log analysis tools or even filtering mechanisms.

to_ids is represented as a JSON boolean. to_ids MUST be present.

#### 2.4.2.6. event_id

event_id represents a human-readable identifier referencing the Event object that the attribute belongs to.

The event_id SHOULD be updated when the event is imported to reflect the newly created event's id on the instance.

event_id is represented as a JSON string. event_id MUST be present.

#### 2.4.2.7. distribution

distribution represents the basic distribution rules of the attribute. The system must adhere to the distribution setting for access control and for dissemination of the attribute.

distribution is represented by a JSON string. distribution MUST be present and be one of the following options:

0
: Your Organisation Only

1
: This Community Only

2
: Connected Communities

3
: All Communities

4
: Sharing Group

5

Inherit Event

#### 2.4.2.8. timestamp

timestamp represents a reference time when the attribute was created or last modified. timestamp is expressed in seconds (decimal) since 1st of January 1970 (Unix timestamp). The time zone MUST be UTC.

timestamp is represented as a JSON string. timestamp MUST be present.

#### 2.4.2.9. comment

comment is a contextual comment field.

comment is represented by a JSON string. comment MAY be present.

#### 2.4.2.10. sharing_group_id

sharing_group_id represents a human-readable identifier referencing a Sharing Group object that defines the distribution of the attribute, if distribution level "4" is set.

sharing_group_id is represented by a JSON string and MUST be present. If a distribution level other than "4" is chosen the sharing_group_id MUST be set to "0".

#### 2.4.2.11. deleted

deleted represents a setting that allows attributes to be revoked. Revoked attributes are not actionable and exist merely to inform other instances of a revocation.

deleted is represented by a JSON boolean. deleted MUST be present.

#### 2.4.2.12. data

data contains the base64 encoded contents of an attachment or a malware sample. For malware samples, the sample MUST be encrypted using a password protected zip archive, with the password being "infected".

data is represented by a JSON string in base64 encoding. data MUST be set for attributes of type malware-sample and attachment.

#### 2.4.2.13. RelatedAttribute

RelatedAttribute is an array of attributes correlating with the current attribute. Each element in the array represents an JSON object which contains an Attribute dictionnary with the external attributes who correlate. Each Attribute MUST include the id, org_id, info and a value. Only the correlations found on the local instance are shown in RelatedAttribute.

RelatedAttribute MAY be present.

#### 2.4.2.14. ShadowAttribute

ShadowAttribute is an array of shadow attributes that serve as proposals by third parties to alter the containing attribute. The structure of a ShadowAttribute is similar to that of an Attribute, which can be accepted or discarded by the event creator. If accepted, the original attribute containing the shadow attribute is removed and the shadow attribute is converted into an attribute.

Each shadow attribute that references an attribute MUST contain the containing attribute's ID in the old_id field and the event's ID in the event_id field.

#### 2.4.2.15. value

value represents the payload of an attribute. The format of the value is dependent on the type of the attribute.

value is represented by a JSON string. value MUST be present.

### 2.5. ShadowAttribute

ShadowAttributes are 3rd party created attributes that either propose to add new information to an event or modify existing information. They are not meant to be actionable until the event creator accepts them - at which point they will be converted into attributes or modify an existing attribute.

They are similar in structure to Attributes but additionally carry a reference to the creator of the ShadowAttribute as well as a revocation flag.

#### 2.5.1. Sample Attribute Object

```
"ShadowAttribute":  {
                        "id": "8",
                        "type": "ip-src",
                        "category": "Network activity",
                        "to_ids": false,
                        "uuid": "57d475f1-da78-4569-89de-1458c0a83869",
                        "event_uuid": "57d475e6-41c4-41ca-b450-145ec0a83869",
                        "event_id": "9",
                        "old_id": "319",
                        "comment": "",
                        "org_id": "1",
                        "proposal_to_delete": false,
                        "value": "5.5.5.5",
                        "deleted": false,
                        "Org": {
                            "id": "1",
                            "name": "MISP",
                            "uuid": "568cce5a-0c80-412b-8fdf-1ffac0a83869"
                        }
                    }
```


### 2.5.2. ShadowAttribute Attributes

#### 2.5.2.1. uuid

uuid represents the Universally Unique IDentifier (UUID) [RFC4122] of the event. The uuid MUST be preserved for any updates or transfer of the same event. UUID version 4 is RECOMMENDED when assigning it to a new event.

uuid is represented as a JSON string. uuid MUST be present.

#### 2.5.2.2. id

id represents the human-readable identifier associated to the event for a specific MISP instance.

id is represented as a JSON string. id SHALL be present.

#### 2.5.2.3. type

type represents the means through which an attribute tries to describe the intent of the attribute creator, using a list of pre-defined attribute types.

type is represented as a JSON string. type MUST be present and it MUST be a valid selection for the chosen category. The list of valid category-type combinations is as follows:

Internal reference
text, link, comment, other

Targeting data
target-user, target-email, target-machine, target-org, target-location, target-external, comment

Antivirus detection
link, comment, text, attachment, other

Payload delivery
md5, sha1, sha224, sha256, sha384, sha512, sha512/224, sha512/256, ssdeep, imphash, authentihash, pehash, tlsh, filename, filename|md5, filename|sha1, filename|sha224, filename|sha256, filename|sha384, filename|sha512, filename|sha512/224, filename|sha512/256, filename|authentihash, filename|ssdeep, filename|tlsh, filename|imphash, filename|pehash, ip-src, ip-dst, hostname, domain, email-src, email-dst, email-subject, email-attachment, url, user-agent, AS, pattern-in-file, pattern-in-traffic, yara, attachment, malware-sample, link, malware-type, comment, text, vulnerability, x509-fingerprint-sha1, other

Artifacts dropped
md5, sha1, sha224, sha256, sha384, sha512, sha512/224, sha512/256, ssdeep, imphash, authentihash, filename, filename|md5, filename|sha1, filename|sha224, filename|sha256, filename|sha384, filename|sha512, filename|sha512/224, filename|sha512/256, filename|authentihash, filename|ssdeep, filename|tlsh, filename|imphash, filename|pehash, regkey, regkey|value, pattern-in-file, pattern-in-memory, pdb, yara, attachment, malware-sample, named pipe, mutex, windows-scheduled-task, windows-service-name, windows-service-displayname, comment, text, x509-fingerprint-sha1, other

Payload installation
md5, sha1, sha224, sha256, sha384, sha512, sha512/224, sha512/256, ssdeep, imphash, authentihash, pehash, tlsh, filename, filename|md5, filename|sha1, filename|sha224, filename|sha256, filename|sha384, filename|sha512, filename|sha512/224, filename|sha512/256, filename|authentihash, filename|ssdeep, filename|tlsh, filename|imphash, filename|pehash, pattern-in-file, pattern-in-traffic, pattern-in-memory, yara, vulnerability, attachment, malware-sample, malware-type, comment, text, x509-fingerprint-sha1, other

Persistence mechanism
filename, regkey, regkey|value, comment, text, other

Network activity
ip-src, ip-dst, hostname, domain, domain|ip, email-dst, url, uri, user-agent, http-method, AS, snort, pattern-in-file, pattern-in-traffic, attachment, comment, text, x509-fingerprint-sha1, other

Payload type
comment, text, other

Attribution
threat-actor, campaign-name, campaign-id, whois-registrant-phone, whois-registrant-email, whois-registrant-name, whois-registrar, whois-creation-date, comment, text, x509-fingerprint-sha1, other

External analysis
md5, sha1, sha256, filename, filename|md5, filename|sha1, filename|sha256, ip-src, ip-dst, hostname, domain, domain|ip, url, user-agent, regkey, regkey|value, AS, snort, pattern-in-file, pattern-in-traffic, pattern-in-memory, vulnerability, attachment, malware-sample, link, comment, text, x509-fingerprint-sha1, other

Financial fraud
btc, iban, bic, bank-account-nr, aba-rtn, bin, cc-number, prtn, comment, text, other

Other
comment, text, other

Attributes are based on the usage within their different communities. Attributes can be extended on a regular basis and this reference document is updated accordingly.

#### 2.5.2.4. category

category represents the intent of what the attribute is describing as selected by the attribute creator, using a list of pre-defined attribute categories.

category is represented as a JSON string. category MUST be present and it MUST be a valid selection for the chosen type. The list of valid category-type combinations is mentioned above.

#### 2.5.2.5. to_ids

to_ids represents whether the Attribute to be created if the ShadowAttribute is accepted is meant to be actionable. Actionable defined attributes that can be used in automated processes as a pattern for detection in Local or Network Intrusion Detection System, log analysis tools or even filtering mechanisms.

to_ids is represented as a JSON boolean. to_ids MUST be present.

#### 2.5.2.6. event_id

event_id represents a human-readable identifier referencing the Event object that the ShadowAttribute belongs to.

The event_id SHOULD be updated when the event is imported to reflect the newly created event's id on the instance.

event_id is represented as a JSON string. event_id MUST be present.

#### 2.5.2.7. old_id

old_id represents a human-readable identifier referencing the Attribute object that the ShadowAttribute belongs to. A ShadowAttribute can this way target an existing Attribute, implying that it is a proposal to modify an existing Attribute, or alternatively it can be a proposal to create a new Attribute for the containing Event.

The old_id SHOULD be updated when the event is imported to reflect the newly created Attribute's id on the instance. Alternatively, if the ShadowAttribute proposes the creation of a new Attribute, it should be set to 0.

old_id is represented as a JSON string. old_id MUST be present.

#### 2.5.2.8. timestamp

timestamp represents a reference time when the attribute was created or last modified. timestamp is expressed in seconds (decimal) since 1st of January 1970 (Unix timestamp). The time zone MUST be UTC.

timestamp is represented as a JSON string. timestamp MUST be present.

#### 2.5.2.9. comment

comment is a contextual comment field.

comment is represented by a JSON string. comment MAY be present.

#### 2.5.2.10. org_id

org_id represents a human-readable identifier referencing the proposal creator's Organisation object.

Whilst attributes can only be created by the event creator organisation, shadow attributes can be created by third parties. org_id tracks the creator organisation.

org_id is represented by a JSON string and MUST be present.

#### 2.5.2.11. proposal_to_delete

proposal_to_delete is a boolean flag that sets whether the shadow attribute proposes to alter an attribute, or whether it proposes to remove it completely.

Accepting a shadow attribute with this flag set will remove the target attribute.

proposal_to_delete is a JSON boolean and it MUST be present. If proposal_to_delete is set to true, old_id MUST NOT be 0.

#### 2.5.2.12. deleted

deleted represents a setting that allows shadow attributes to be revoked. Revoked shadow attributes only serve to inform other instances that the shadow attribute is no longer active.

deleted is represented by a JSON boolean. deleted SHOULD be present.

#### 2.5.2.13. data

data contains the base64 encoded contents of an attachment or a malware sample. For malware samples, the sample MUST be encrypted using a password protected zip archive, with the password being "infected".

data is represented by a JSON string in base64 encoding. data MUST be set for shadow attributes of type malware-sample and attachment.

### 2.5.3. Org

An Org object is composed of an uuid, name and id.

The uuid represents the Universally Unique IDentifier (UUID) [RFC4122] of the organization. The organization UUID is globally assigned to an organization and SHALL be kept overtime.

The name is a readable description of the organization and SHOULD be present. The id is a human-readable identifier generated by the instance and used as reference in the event.

uuid, name and id are represented as a JSON string. uuid, name and id MUST be present.

#### 2.5.3.1. Sample Org Object

```
"Org": {
        "id": "2",
        "name": "CIRCL",
        "uuid": "55f6ea5e-2c60-40e5-964f-47a8950d210f"
       }
```

#### 2.5.3.2. value

value represents the payload of an attribute. The format of the value is dependent on the type of the attribute.

value is represented by a JSON string. value MUST be present.

### 2.6. Tag

A Tag is a simple method to classify an event with a simple tag name. The tag name can be freely chosen. The tag name can be also chosen from a fixed machine-tag vocabulary called MISP taxonomies[[MISP-T]]. A Tag is represented as a JSON array where each element describes each tag associated. A Tag array SHALL be, at least, at Event level. A tag element is described with a name, id, colour and exportable flag.

exportable represents a setting if the tag is kept local or exportable to other MISP instances. exportable is represented by a JSON boolean. id is a human-readable identifier that references the tag on the local instance. colour represents an RGB value of the tag.

name MUST be present. colour, id and exportable SHALL be present.

#### 2.6.1. Sample Tag

```
"Tag": [{
        "exportable": true,
        "colour": "#ffffff",
        "name": "tlp:white",
        "id": "2" }]
```

## 3. Manifest

MISP events can be shared over an HTTP repository, a file package or USB key. A manifest file is used to provide an index of MISP events

allowing to only fetch the recently updated files without the need to parse each json file.

### 3.1. Format

A manifest file is a simple JSON file named manifest.json in a directory where the MISP events are located. Each MISP event is a file located in the same directory with the event uuid as filename with the json extension.

The manifest format is a JSON object composed of a dictionary where the field is the uuid of the event.

Each uuid is composed of a JSON object with the following fields which came from the original event referenced by the same uuid:

- info (MUST)
- Orgc object (MUST)
- analysis (SHALL)
- timestamp (MUST)
- date (MUST)
- threat_level_id (SHALL)

In addition to the fields originating from the event, the following fields can be added:

- integrity:sha256 represents the SHA256 value in hexadecimal representation of the associated MISP event file to ensure integrity of the file. (SHOULD)
- integrity:pgp represents a detached PGP signature [RFC4880] of the associated MISP event file to ensure integrity of the file. (SHOULD)

If a detached PGP signature is used for each MISP event, a detached PGP signature is a MUST to ensure integrity of the manifest file. A detached PGP signature for a manifest file is a manifest.json.pgp file containing the PGP signature.

### 3.1.1. Sample Manifest

```
{
  "57c6ac4c-c60c-4f79-a38f-b666950d210f": {
    "info": "Malspam 2016-08-31 (.wsf in .zip) - campaign: Photo",
    "Orgc": {
      "id": "2",
      "name": "CIRCL"
    },
    "analysis": "0",
    "Tag": [
      {
        "colour": "#3d7a00",
        "name": "circl:incident-classification=\"malware\""
      },
      {
        "colour": "#ffffff",
        "name": "tlp:white"
      }
    ],
    "timestamp": "1472638251",
    "date": "2016-08-31",
    "threat_level_id": "3"
  },
  "5720accd-dd28-45f8-80e5-4605950d210f": {
    "info": "Malspam 2016-04-27 - Locky",
    "Orgc": {
      "id": "2",
      "name": "CIRCL"
    },
    "analysis": "2",
    "Tag": [
      {
        "colour": "#ffffff",
        "name": "tlp:white"
      },
      {
        "colour": "#3d7a00",
        "name": "circl:incident-classification=\"malware\""
      },
      {
        "colour": "#2c4f00",
        "name": "malware_classification:malware-category=\"Ransomware\""
      }
    ],
    "timestamp": "1461764231",
    "date": "2016-04-27",
    "threat_level_id": "3"
  }
}
```

## 4. Implementation

MISP format is implemented by different software including the MISP threat sharing platform and libraries like PyMISP [MISP-P]. Implementations use the format as an export/import mechanism, staging transport format or synchronisation format as used in the MISP core platform. MISP format doesn't impose any restriction on the data representation of the format in data-structure of other implementations.

## 5. Security Considerations

MISP events might contain sensitive or confidential information. Adequate access control and encryption measures shall be implemented to ensure the confidentiality of the MISP events.

Adversaries might include malicious content in MISP events and attributes. Implementation MUST consider the input of malicious inputs beside the standard threat information that might already include malicious intended inputs.

## 6. Acknowledgements

The authors wish to thank all the MISP community to support the creation of open standards in threat intelligence sharing.

## 7. Sample MISP file

## 8. References

### 8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, DOI 10.17487/RFC4122, July 2005, <http://www.rfc-editor.org/info/rfc4122>.

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, DOI 10.17487/RFC4627, July 2006, <http://www.rfc-editor.org/info/rfc4627>.

[RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, DOI 10.17487/RFC4880, November 2007, <http://www.rfc-editor.org/info/rfc4880>.

### 8.2. Informative References

[MISP-P] MISP, , "MISP Project - Malware Information Sharing Platform and Threat Sharing", <https://github.com/MISP>.

[MISP-T] MISP, , "MISP Taxonomies - shared and common vocabularies of tags", <https://github.com/MISP/misp-taxonomies>.

## Authors' Addresses

Alexandre Dulaunoy
Computer Incident Response Center Luxembourg
41, avenue de la gare
Luxembourg L-1611
Luxembourg

Phone: +352 247 88444
Email: alexandre.dulaunoy@circl.lu

Andras Iklody
Computer Incident Response Center Luxembourg
41, avenue de la gare
Luxembourg L-1611
Luxembourg

Phone: +352 247 88444
Email: andras.iklody@circl.lu