Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 18, 2019

A. Dulaunoy
A. Iklody
CIRCL

October 15, 2018

MISP object template format
draft-dulaunoy-misp-object-template-format-02

Abstract

This document describes the MISP object template format which describes a simple JSON format to represent the various templates used to construct MISP objects. A public directory of common vocabularies MISP object templates [MISP-O] is available and relies on the MISP object reference format.

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 https://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, 2019.

Table of Contents

1. Introduction 2
1.1. Conventions and Terminology 2
2. Format 3
2.1. Overview 3
2.1.1. Object Template 3
2.1.2. attributes 4
2.1.3. Sample Object Template object 6
2.1.4. Object Relationships 9
3. Directory 10
4. Acknowledgements 10
5. References 10
5.1. Normative References 10
5.2. Informative References 10
Authors' Addresses 11

1. Introduction

Due to the increased maturity of threat information sharing, the need arose for more complex and exhaustive data-points to be shared across the various sharing communities. MISP's information sharing in general relied on a flat structure of attributes contained within an event, where attributes served as atomic secluded data-points with some commonalities as defined by the encapsulating event. However, this flat structure restricted the use of more diverse and complex data-points described by a list of atomic values, a problem solved by the MISP object structure.

MISP objects combine a list of attributes to represent a singular object with various facets. In order to bootstrap the object creation process and to maintain uniformity among objects describing similar data-points, the MISP object template format serves as a reusable and share-able blueprint format.

MISP object templates also include a vocabulary to describe the various inter object and object to attribute relationships and are leveraged by MISP object references.

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

MISP object templates are composed of the MISP object template (MUST) structure itself and a list of MISP object template elements (SHOULD) describing the list of possible attributes belonging to the resulting object, along with their context and settings.

MISP object templates themselves consist of a name (MUST), a meta-category (MUST) and a description (SHOULD). They are identified by a uuid (MUST) and a version (MUST). For any updates or transfer of the same object reference. UUID version 4 is RECOMMENDED when assigning it to a new object reference. The list of requirements when it comes to the contained MISP object template elements is defined in the requirements field (OPTIONAL).

MISP object template elements consist of an object_relation (MUST), a type (MUST), an object_template_id (SHOULD), a ui_priority (SHOULD), a list of categories (MAY), a list of sane_default values (MAY) or a values_list (MAY).

2.1. Overview

The MISP object template format uses the JSON [RFC4627] format. Each template is represented as a JSON object with meta information including the following fields: uuid, requiredOneOf, description, version, meta-category, name.

2.1.1. Object Template

2.1.1.1. uuid

uuid represents the Universally Unique IDentifier (UUID) [RFC4122] of the object template. The uuid MUST be preserved for to keep consistency of the templates across instances. UUID version 4 is RECOMMENDED when assigning it to a new object template.

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

2.1.1.2. requiredOneOf

requiredOneOf is represented as a JSON list and contains a list of attribute relationships of which one must be present in the object to be created based on the given template. The requiredOneOf field MAY be present.

2.1.1.3. required

required is represented as a JSON list and contains a list of attribute relationships of which all must be present in the object to be created based on the given template. The required field MAY be present.

2.1.1.4. description

description is represented as a JSON string and contains the assigned meaning given to objects created using this template. The description field MUST be present.

2.1.1.5. version

version represents a numeric incrementing version of the object template. It is used to associate the object to the correct version of the template and together with the uuid field forms an association to the correct template type and version.

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

2.1.1.6. meta-category

meta-category represents the sub-category of objects that the given object template belongs to. meta-categories are not tied to a fixed list of options but can be created on the fly.

meta-category is represented as a JSON string. meta-category MUST be present.

2.1.1.7. name

name represents the human-readable name of the objects created using the given template, describing the intent of the object package.

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

2.1.2. attributes

attributes is represented as a JSON list and contains a list of template elements used as a template for creating the individual attributes within the object that is to be created with the object.

attributes is represented as a JSON list. attributes MUST be present.

2.1.2.1. description

description is represented as a JSON string and contains the description of the given attribute in the context of the object with the given relationship. The description field MUST be present.

2.1.2.2. ui-priority

ui-priority is represented by a numeric values in JSON string format and is meant to provide a priority for the given element in the object template visualisation. The ui-priority MAY be present.

2.1.2.3. misp-attribute

misp-attribute is represented by a JSON string or a JSON object with a list of values. The value(s) are taken from the pool of types defined by the MISP core format's Attribute Object's type list. type can contain a JSON object with a list of suggested value alternatives encapsulated in a list within a sane_default key or a list of enforced value alternatives encapsulated in a list_values key.

The misp-attribute field MUST be present.

2.1.2.4. disable_correlation

disable_correlation is represented by a JSON boolean. The disable_correlation field flags the attribute(s) created by the given object template element to be marked as non correlating.

The misp-attribute field MAY be present.

2.1.2.5. categories

categories is represented by a JSON list containing one or several valid options from the list of verbs valid for the category field in the Attribute object within the MISP core format.

The categories field MAY be present.

2.1.2.6. multiple

multiple is represented by a JSON boolean value. It marks the MISP object template element as a multiple input field, allowing for several attributes to be created by the element within the same object.

The multiple field MAY be present.

2.1.2.7. sane_default

sane_default is represented by a JSON list containing one or several recommended/sane values for an attribute. sane_default is mutually exclusive with values_list.

The sane_default field MAY be present.

2.1.2.8. values_list

values_list is represented by a JSON List containing one or several of fixed values for an attribute. values_list is mutually exclusive with sane_default.

The value_list field MAY be present.

2.1.3. Sample Object Template object

The MISP object template directory is publicly available [MISP-O] in a git repository and contains more than 60 object templates. As illustration, two sample objects templates are included.

2.1.3.1. credit-card object template

```
{
  "requiredOneOf": [
    "cc-number"
  ],
  "attributes": {
    "version": {
      "description": "Version of the card.",
      "ui-priority": 0,
      "misp-attribute": "text"
    },
    "comment": {
      "description": "A description of the card.",
      "ui-priority": 0,
      "misp-attribute": "comment"
    },
    "card-security-code": {
      "description": "Card security code (CSC, CVD, CVV, CVC and SPC) as
embossed or printed on the card.",
      "ui-priority": 0,
      "misp-attribute": "text"
    },
    "name": {
      "description": "Name of the card owner.",
      "ui-priority": 0,
      "misp-attribute": "text"
    },
    "issued": {
      "description": "Initial date of validity or issued date.",
      "ui-priority": 0,
      "misp-attribute": "datetime"
    },
    "expiration": {
      "description": "Maximum date of validity",
      "ui-priority": 0,
      "misp-attribute": "datetime"
    },
    "cc-number": {
      "description": "credit-card number as encoded on the card.",
      "ui-priority": 0,
      "misp-attribute": "cc-number"
    }
  },
  "version": 2,
  "description": "A payment card like credit card, debit card or any similar
cards which can be used for financial transactions.",
  "meta-category": "financial",
  "uuid": "2b9c57aa-daba-4330-a738-56f18743b0c7",
  "name": "credit-card"
```

```
}
```

2.1.3.2. credential object template

```
{
  "requiredOneOf": [
    "password"
  ],
  "attributes": {
    "text": {
      "description": "A description of the credential(s)",
      "disable_correlation": true,
      "ui-priority": 1,
      "misp-attribute": "text"
    },
    "username": {
      "description": "Username related to the password(s)",
      "ui-priority": 1,
      "misp-attribute": "text"
    },
    "password": {
      "description": "Password",
      "multiple": true,
      "ui-priority": 1,
      "misp-attribute": "text"
    },
    "type": {
      "description": "Type of password(s)",
      "ui-priority": 1,
      "misp-attribute": "text",
      "values_list": [
        "password",
        "api-key",
        "encryption-key",
        "unknown"
      ]
    },
    "origin": {
      "description": "Origin of the credential(s)",
      "ui-priority": 1,
      "misp-attribute": "text",
      "sane_default": [
        "bruteforce-scanning",
        "malware-analysis",
        "memory-analysis",
        "network-analysis",
        "leak",
        "unknown"
      ]
    },
```

```
    "format": {
      "description": "Format of the password(s)",
      "ui-priority": 1,
      "misp-attribute": "text",
      "values_list": [
        "clear-text",
        "hashed",
        "encrypted",
        "unknown"
      ]
    },
    "notification": {
      "description": "Mention of any notification(s) towards the potential
owner(s) of the credential(s)",
      "ui-priority": 1,
      "misp-attribute": "text",
      "multiple": true,
      "values_list": [
        "victim-notified",
        "service-notified",
        "none"
      ]
    }
  },
  "version": 2,
  "description": "Credential describes one or more credential(s) including
password(s), api key(s) or decryption key(s).",
  "meta-category": "misc",
  "uuid": "a27e98c9-9b0e-414c-8076-d201e039ca09",
  "name": "credential"
}
```


2.1.4. Object Relationships

2.1.4.1. name

name represents the human-readable relationship type which can be used when creating MISP object relations.

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

2.1.4.2. description

description is represented as a JSON string and contains the description of the object relationship type. The description field MUST be present.

2.1.4.3. format

format is represented by a JSON list containing a list of formats that the relationship type is valid for and can be mapped to. The format field MUST be present.

3. Directory

The MISP object template directory is publicly available [MISP-O] in a git repository. The repository contains an objects directory, which contains a directory per object type, containing a file named definition.json which contains the definition of the object template in the above described format.

A relationships directory is also included, containing a definition.json file which contains a list of MISP object relation definitions. There are more than 90 existing templates object documented in [MISP-O-DOC].

4. Acknowledgements

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

5. References

5.1. 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, <https://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, <https://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, <https://www.rfc-editor.org/info/rfc4627>.

5.2. Informative References

[MISP-O] MISP, "MISP Objects - shared and common object templates", <https://github.com/MISP/misp-objects>.

[MISP-O-DOC]
"MISP objects directory", 2018,
<https://www.misp-project.org/objects.html>.

Authors' Addresses

Alexandre Dulaunoy
Computer Incident Response Center Luxembourg
16, bd d'Avranches
Luxembourg L-1611
Luxembourg

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

Andras Iklody
Computer Incident Response Center Luxembourg
16, bd d'Avranches
Luxembourg L-1611
Luxembourg

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