Workgroup: Network Working Group
Internet-Draft: draft-lehmann-idmefv2-00
Obsoletes: 4765 (if approved)
Published: 15 October 2022
Intended Status: Standards Track
Expires: 18 April 2023

Authors: G. Lehmann          T. Andrejak   F. Poirotte
         Telecom Sud Paris   CS GROUP      CS GROUP

# The Incident Detection Message Exchange Format version 2 (IDMEFv2)

## Abstract

The Incident Detection Message Exchange Format version 2 (IDMEFv2) provides a way to describe any incidents detected on cyber and/or physical infrastructures.

The format is agnostic so it can be used in standalone or combined cyber (SIEM), physical (PSIM) and availability (NMS) monitoring systems. IDMEFv2 can also be used to describe cyber and physical potential threats (CTI/PTI).

IDMEFv2 improves situational awareness by facilitating correlation of multiple types of events using the same base format thus enabling efficient detection of complex and combined cyber and physical attacks on critical infrastructures.

If approved this draft will obsolete RFC4765.

## Status of This Memo

## Copyright Notice

## Table of Contents

## 1. Introduction

Today's threats are a result of hybrid attacks targeting both physical and cyber assets. The adoption and integration of Internet of Things (IoT) and Industrial Internet of Things (IIoT) devices have led to an increasingly interconnected mesh of cyber-physical systems (CPS), which expands the attack surface and blurs the once clear functions of cybersecurity and physical security. Meanwhile, efforts to build cyber resilience and accelerate the adoption of advanced technologies can also introduce or exacerbate security risks in this evolving threat landscape.

In the meantime, although security is often presented as the Confidentiality-Integrity-Availability triad, performance and availability management systems are still run independently from security management systems making global correlation difficult.

The Incident Detection Message Exchange Format (IDMEF) is intended to be a standard data format that incident detection systems can use to report alerts about events that they deem noticeable. The format enables interoperability among commercial, open source, and research systems, allowing users to mix-and-match the deployment of these systems according to their strong and weak points to obtain an optimal implementation.

The Incident Detection Message Exchange Format is a format for representing different types of events:

*Cyber-security events (e.g. authentication failure/success, virus/malware detection, bruteforce/scan detection, etc.)

*Physical security events (e.g. intrusion detection, object detection, face or activity recognition, fire/smoke/noise/rain detection, etc.)

*Availability/observability/performance events (e.g. system failure, service malfunction, performance decrease, etc.)

*Natural hazards events (e.g. wildfires, avalanches, droughts, earthquakes, etc.)

```
                    +-----------------------------+    +----------+
                    |      "Universal" SI(E)M     |<---| PTI/CTI  |
                    +-----------------------------+    +----------+
                        |          |         |
                     +------+   +-----+   +------+
    Managers         | PSIM |   | NMS |   | SIEM |
                     +------+   +-----+   +------+
                        |          |         |
                    +--------+ +----------+ +-----+
Detectors/Sensors   |Physical| |Monitoring| |Cyber|
                    +--------+ +----------+ +-----+
                        |           |          |
                   +--------------------------------+
                   |   Critical Infrastructure      |
                   +--------------------------------+
```

Figure 1: IDMEF Use Architecture

IDMEF improves situational awareness by enabling correlation of multiple types of events using the same base format.

This document defines a model for the purpose of describing these events. It also defines serialization methods so that such messages can be exchanged between Computer Security Incident Response Teams (CSIRTs) or those responsible for security incident handling for service providers (SPs). The defined serializations make it easy for CSIRTs to exchange data in a way that is both easy and secure for machines to parse.

### 1.1. Issues and limitations in RFC 4765

The original IDMEF (version 1) RFC [RFC4765] was specifically designed to describe alerts related to cyber intrusions. As such, its data model makes it hard to describe other types of (cyber) incidents.

IDMEF v1 defines many classes and attributes, adding a lot of complexity. Some constructs (e.g. use of recursive Analyzer instances, unlimited usage of the Linkage class, etc.) make the implementators' job hard.

RFC 4765 uses the Extensible Markup Language (XML) to describe IDMEF classes and attributes, using an XML Document Type Definition. It does not specify however if the XML representation of IDMEF messages must be used when exchanging messages with other systems/tools. In practice, this lack of a requirement means that competiting implementations may use incompatible protocols to do so.

In addition, XML suffers from a number of specific flaws which can be easy to overlook and difficult to address depending on the tooling used:

- XML External Entity (XXE) vulnerabilities may be used to include external (potentially remote) content inside the XML document during processing. This may impact the integrity of the IDMEF messages, result in unintentional information disclosure, etc.

- XInclude processing may result in the inclusion of potentially remote content, similar to the XXE vulnerability above.

- XML Entity bombs like the so-called "Billion laughs" attack can result in a denial of service against IDMEF processors by exhausting the system's CPU and memory resources.

As such, the use of XML as an exchange format can be problematic.

### 1.2. Changes from RFC 4765

Several changes have been made compared to the original IDMEF v1 RFC [RFC4765]:

- The first version of IDMEF (i.e. the Intrusion Detection Message Exchange Format) was specifically designed to describe only alerts related to cyber intrusions. This document redefines IDMEF as the "Incident Detection Message Exchange Format".

  This change is made to include other sources of incidents that may impact a company's security. For instance, the failure of a service may be due to a physical intrusion followed by sabotage, some hardware failure, a natural disaster, etc., or to a combination of several types of incidents.

As an intrusion is only part of the incidents that IDMEF v2 intends to describe, it makes sense to allow IDMEF to address a broader scope. In addition, this means that this documents is semantically backward compatible with the former RFC.

- Simplicity and ease of adoption have been preferred over completeness and complexity. As a result of this simplification, the number of classes and attributes has been reduced. Moreover, the model has been reworked to limit the depth of classes to two levels.

- A "Sensor" class has been added to help distinguish detection systems made of a separate detector and analyzer (e.g. a camera recording a video feed and the backend server/software component analyzing this feed).

- An "Attachment" class has been added to attach additional data to the alert (e.g. a video clip, a malware sample, etc.).

- The "Observable" and "Vector" classes have been added to describe the attack vectors and observable effects/measurements related to the incident.

- The Hearbeat class has been abandonned.

### 1.3. About the JSON serialization method

Although the IDMEF data model strives to be independent from any particular representation, such a serialization is necessary if IDMEF is to be used as an exchange format. Moreover, an interoperable serialization scheme is required for compatibility reasons.

This document describes a serialization method for IDMEF messages based on the JavaScript Object Notation [RFC8259]. This choice is motivated by the following factors:

- The format is already largely used inside the cybersecurity community, e.g. to replace the syslog format for log shopping. It thus lowers the level of entry for implementors.

- JSON is often seen as a simpler format compared to XML, from both an implementor's and user's point of view. Because of the way XML works, XML documents are usually larger than JSON ones when representing the same content, due for example to the use of namespaces and the repetition of the elements' tag name inside the markup.

- An effort has been made to make IDMEF useable from end to end, i.e. from the incident detectors to the operator. IDMEF messages must therefore be easy to store in a database, especially NoSQL databases which are often used to store very large amounts of data. JSON is a good format for native NoSQL storage.

In contrast, the authors acknowledge that:

- JSON may suffer from issues of its own. For instance, string processing may require additional normalization steps (e.g. when comparing two JSON strings). and two JSON parsers may handle duplicate members inside a JSON object differently. These concerns are largely covered in [RFC8259] and in this document's Security Considerations (Section 6).

- Other formats similar to JSON could also fit this role (e.g. YAML, TOML). Those formats are less widely used by incident management tools and operators. They may also introduce vulnerabilities and incompatibilities of their own (e.g. there are multiple versions of YAML, a YAML document may call implementation-specific functions used "tags", etc.). In addition, most of those formats focus on human-readability, while for the purpose of IDMEF, the main objectives are performance and security.

### 1.4. Relationship between IDMEFv2 and other event/incident formats

IDMEFv1 : IDMEFv2 (Incident Detection) replaces and obsoletes IDMEFv1 (Intrusion Detection) by covering a wider spectrum.

IODEFv2 : IDMEFv2 helps detect incident that will after be fully described with IODEFv2. IDMEF is used upstream IODEFv2.

Syslog : IDMEFv2 can be used as an alternative to syslog for detectors needing to log detailed information of an event and/or an incident.

SNMP : SNMP polls information from devices which is then compared to thresholds to detect incident. IDMEFv2 can be used when incident is detected downstream of SNMP. IDMEFv2 can have a similar role as SNMP Traps.

STIX : IDMEFv2 can help gathering information for creation of CTI.

SIEM propriatory formats (CEF, LEEF, ECS, CIM, ...) : By covering physical and monitoring incident type, IDMEFv2 offers a wider spectrum than those formats. Gateways between IDMEFv2 and those formats can be developped.

# 2. Terminology

## 2.1. Keywords

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 [RFC2119].

## 2.2. Normative sections

Implementations of IDMEFv2 are REQUIRED to fully implement:

- The data types defined in Section 3

- The data model defined in Section 4

- The JavaScript Object Notation (JSON) serialization method Section 4.10.

## 2.3. Concepts related to event processing

### 2.3.1. Event

An event is something that triggered a notice. Any incident starts off as an event or a combination of events, but not all events result in an incident. An event need not be an indication of wrongdoing. E.g. someone successfully logging in or entering a building is an event.

### 2.3.2. Incident

An incident is an event that compromises or has a significant probability of compromising at least one of the organization's security criteria such as Confidentiality, Integrity or Availability. An incident may affect a production tool, personnel, etc. It may be logical, physical or organizational in nature. Last but not least, an incident may be caused on purpose or by accident.

### 2.3.3. Alert

An alert is a notification/message that a particular event/incident (or series of events/incidents) has occurred.

### 2.3.4. Attack

An attack is an attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of a cyber or physical asset. An attack is one or many kinds of incidents.

#### 2.3.5. Correlation

Correlation is the identification of relationships between two or more events.

#### 2.3.6. Aggregation

Aggregation is the consolidation of similar events into a single event.

## 3. The IDMEF Data Types

Each object inside the IDMEF data model has an associated data type. This type may be used to validate the content of incoming IDMEF messages.

### 3.1. Classes

The classes are meant to group related attributes together. Some of the classes may be instanciated multiple times (e.g. Source, Target, etc.) while others may only appear once in an IDMEF message (e.g. Analyzer).

### 3.2. Numbers

#### 3.2.1. Integers

Integers inside the IDMEF data model are expressed using the following ABNF [RFC5234] grammar:

```
integer         =  *1minus int
int             =  zero / ( digit1-9 *DIGIT )
minus           =  %x2D                        ; -
zero            =  %0x30                       ; 0
digit1-9        =  %x31-39                     ; 1-9
```

E.g. 123.

Such values are indicated with the "INT" type annotation in the model.

#### 3.2.2. Floating-point values

Floating-point values inside the IDMEF data model are expressed using the following ABNF grammar:

```
float           =  integer *1frac
frac            =  decimal-point 1*DIGIT
decimal-point   =  %x2E                             ; .
```

This grammar reuses some of the production rules listed in Section 3.2.1.

E.g. 12.34.

Such values are indicated with the "FLOAT" type annotation in the model.

### 3.3. Strings

Strings are series of characters from the [UNICODE] standard and are used to represent a text.

For readability, this document uses quotes (") to delimit strings, but please note that these quotes are not syntactically part of the actual strings.

E.g. "Hello world".

Some of the strings used in the IDMEFv2 data model follow a stricter syntax. These are included below for completeness.

Such values are indicated with the "STRING" type annotation in the model.

#### 3.3.1. Enumerations

Enumerations are special strings used when valid values for an IDMEF attribute are restricted to those present in a predefined list.

Such values are indicated with the "ENUM" type annotation in the model.

#### 3.3.2. Timestamps

Timestamps are used to indicate a specific moment in time. The timestamps used in the IDMEF data model follow the syntax defined by the "date-time" production rule of the grammar in [RFC3339] ch 5.6.

E.g. "1985-04-12T23:59:59.52Z" represents a moment just before April 5th, 1985 in Coordinated Universal Time (UTC).

Such values are indicated with the "TIMESTAMP" type annotation in the model.

### 3.3.3. Geographical Locations

Some attributes inside the IDMEF data model may refer to geographical locations using a set of coordinates. The reference system for all geographical coordinates is a geographic coordinate reference system, using the World Geodetic System 1984 [WGS84]. The reference system used is the same as for the Global Positioning System (GPS).

The format for such values can be either "latitude,longitude" or "latitude,longitude,altitude". Each of these coordinates is represented as a floating-point value. The latitude and longitude are expressed in degrees while the altitude is expressed in meters.

E.g. "48.8584,2.2945,276.13" matches the (3-dimensional) geographical location for the top floor or the Eiffel Tower located in Paris, France, while "48.8584,2.2945" matches the same location in two dimensions (with the altitude removed).

Such values are indicated with the "GEOLOC" type annotation in the model.

### 3.3.4. UNECE Location Codes (UN/LOCODE)

Some attributes inside the IDMEF data model may refer to geographical locations using Locations Codes. These codes can be assimilated to an enumeration, where the list of possible values is defined in the United Nations Economic Commission for Europe (UNECE) Codes for Trade [UN-LOCODE].

E.g. "FR PAR" is the Location Code for the city of Paris, France.

Such values are indicated with the "UNLOCODE" type annotation in the model.

### 3.3.5. Uniform Resource Identifiers (URIs)

The IDMEF data model uses Uniform Resource Identifiers (URIs), as defined in [RFC3986], when referring to external resources. Unless otherwise specified, either a Uniform Resource Location (URL) or a Uniform Resource Name (URN) may be used where a URI is expected.

E.g. both "https://example.com/resource" and "urn:myapp:resource" are valid Uniform Resource Identifiers.

Such values are indicated with the "URI" type annotation in the model.

### 3.3.6. IP Addresses

IP addresses inside the IDMEF data model are expressed as strings using the traditionnal dotted-decimal notation for IPv4 addresses (defined by the "dotnum" production rule in the grammar in [RFC5321]), while IPv6 addresses are expressed using the text representation defined in [RFC4291] ch 2.2.

E.g. "192.0.2.1" represents a valid IPv4 address, while "::1/128" represents a valid IPv6 address.

It is RECOMMENDED that implementations follow the recommendations for IPv6 text representation stated in [RFC5952].

Such values are indicated with the "IP" type annotation in the model.

### 3.3.7. E-mail addresses

E-mail addresses inside the IDMEF data model are expressed as strings using the address specification syntax defined in [RFC5322] ch 3.4.1.

E.g. "root@example.com".

Such values are indicated with the "EMAIL" type annotation in the model.

### 3.3.8. Attachment and Observable names

Attachments and Observables inside the IDMEF data model are identified using a unique name, composed of a string whose character set is limited to the ASCII letters (A-Z a-z) and digits (0-9).

E.g. "state" is a valid name for an attachment or an observable.

The constraint on name unicity is enforced per class. That is, it is perfectly okay for an attachment and an observable to use the same name, but it is not possible for two attachments or two observables to share the same name.

Such values are indicated with the "ID" type annotation in the model.

### 3.3.9. Media types

Media types are used in the IDMEF data model to describe an attachment's content. The syntax for such values is defined in [RFC2046].

IANA keeps a list of all currently registered media types in the Media Types registry .

E.g. "application/xml" or "text/plain; charset=utf-8".

Such values are indicated with the "MEDIATYPE" type annotation in the model.

### 3.3.10. Universally Unique IDentifiers (UUIDs)

Universally Unique Identifiers (UUIDs) are used to uniquely identify IDMEF messages. It is also possible for an IDMEF message to reference other IDMEF messages using their UUIDs. The syntax for UUIDs is defined in [RFC4122].

To limit the risk of UUID collisions, implementors SHOULD NOT generate version 4 UUIDs (randomly or pseudo-randomly generated UUIDs).

E.g. "ba2e4ef4-8719-42bb-a712-d6e8871c5c5a".

UUIDs are case-insensitive when used in comparisons.

Such values are indicated with the "UUID" type annotation in the model.

### 3.3.11. Protocol Names

Such values are indicated with the "PROTOCOL" type annotation in the model.

### 3.3.12. IDMEF Paths

This document defines a way to represent the path to every possible attribute inside an IDMEF message. For conciseness, the top-level "Alert" class is omitted from the path.

This representation can be used in contexts where the path to an IDMEF attribute is expected. An example of such usage can be seen in the definition of the "AggrCondition" attribute inside the Alert class (Section 4.2).

The syntax for these IDMEF paths is expressed in the following ABNF grammar:

```
class-name      =  "Analyzer" / "Sensor" / "Source" / "Target" /
                   "Vector" / "Observable" / "Attachment"
attribute-name  =  1*ALPHA
class-reference =  class-name "."
num             =  *1"-" 1*DIGIT
list-index      =  "(" num ")"
path            =  *1class-reference attribute-name *1list-index
```

Valid attribute names are limited to those defined for the specified class-reference (or in the top-level "Alert" class if class-reference is omitted).

For example, the following path refers to the "CeaseTime" attribute of the top-level "Alert" class: "CeaseTime".

Likewise, the following path refers to the "Name" attribute of the "Analyzer" class: "Analyzer.Name".

For attributes defined as lists (see Section 3.4), the path may include the (0-based) index for an entry inside the list. The index defaults to 0 if omitted. This means that several (valid) representations may be used to reference the same IDMEF attribute when list attributes are involved.

For example, both of the following paths refer to the IP address of the first source associated with an IDMEF message:

```
Source.IP
Source(0).IP
```

Compatible implementations MUST reject paths that reference an unknown class, an unknown attribute, or use a list-index for an IDMEF field which is not defined as a list.

A compatible implementation MUST also normalize paths before comparing them (e.g. by stripping the text "(0)" from paths referring to list attributes).

### 3.3.13.  Hashes

Hashes are sometimes used inside the data model to protect the integrity (and optionally, authenticity) of attachments.

The syntax for these values is "function:hash_result", where "function" refers to one of the hashing function names listed in and "hash_result" contains the hexadecimal notation for the hash result obtained by calling the specified hash function on the input value.

In the context of IDMEF, either a keyless or keyed hash function may be used to process the raw input value.

E.g.
"sha256:a02735ed8b10ad432d557bd4849c0dac3b23d64706e0618716d6df2def33
8374"

Hashes are case-insensitive when used in comparisons.

Such values are indicated with the "HASH" type annotation in the model.

### 3.4. Lists

Some attributes of the IDMEF data model accept ordered lists of values.

Such ordered lists are indicated with the "X[]" type annotation in the model. where "X" refers to one of the data types defined in Section 3. For example, "ENUM[]" refers to an ordered list of enumeration values.

## 4. The IDMEF Data Model

In this section, the individual components of the IDMEF data model will be discussed in detail. For each class, the semantics will be described.

### 4.1. Overview

An IDMEF message is composed of an instance of the Alert class (Section 4.2) representing the overall properties of the message. It also contains exactly one instance of the Analyzer class (Section 4.3) and zero or more instances of the Sensor class (Section 4.4).

The message may also describe various aspects of an incident using the Source (Section 4.5), Target (Section 4.6) and Vector (Section 4.7) classes.

Last but not least, it may also include zero or more instances of the Attachment class (Section 4.8), e.g. captured files or network packets related to the event, as well as zero or more instances of the Observable class (Section 4.9) containing information that may help in understanding and analyzing the event, such as a description of running processes at the time the event occurred, a description of the targeted machine's configuration, etc.

The relationship between the main Alert class and other classes of the data model is shown in Figure 2 (attributes are omitted for clarity).

```
+-------+            +-------------
| Alert |<>----------|  Analyzer  |
+-------+            +------------+
|       |       0..* +------------+
|       |<>----------|   Sensor   |
|       |            +------------+
|       |       0..* +------------+
|       |<>----------|   Source   |
|       |            +------------+
|       |       0..* +------------+
|       |<>----------|   Target   |
|       |            +------------+
|       |       0..* +------------+
|       |<>----------|   Vector   |
|       |            +------------+
|       |       0..* +------------+
|       |<>----------| Observable |
|       |            +------------+
|       |       0..* +------------+
|       |<>----------| Attachment |
+-------+            +------------+
```

Figure 2: IDMEFv2 Classes

It is important to note that the data model does not specify how an alert should be categorized or identified. For example, an attacker scanning a network for machines listening on a specific port may be identified by one analyzer as a single attack against multiple targets, while another analyzer may identify it as multiple attacks from a single source. However, once an analyzer has determined the type of alert it plans on sending, the data model dictates how that alert should be formatted.

## 4.2. The Alert Class

The Alert class contains high level information about the event that triggered the alert.

```
+---------------------------+
|           Alert           |
+---------------------------+
| STRING      Version       |
| UUID        ID            |
| STRING      Entity        |
| ENUM[]      Category      |
| ENUM        Cause         |
| STRING      Description   |
| ENUM        Status        |
| ENUM        Severity      |
| FLOAT       Confidence    |
| STRING      Note          |
| TIMESTAMP   CreateTime    |
| TIMESTAMP   StartTime     |
| TIMESTAMP   CeaseTime     |
| TIMESTAMP   DeleteTime    |
| STRING[]    AltNames      |
| STRING[]    AltCategory   |
| URI[]       Ref           |
| UUID[]      CorrelID      |
| CONDITION[] AggrCondition |
| UUID[]      PredID        |
| UUID[]      RelID         |
+---------------------------+
```

Figure 3: The Alert class

The aggregate classes that make up Alert are:

**Analyzer**
Exactly one. An instance of the Analyzer class (Section 4.3) that describes the tool/device responsible for the analysis that resulted in the alert being sent.

**Sensor**
Zero or more. Instances of the Sensor class (Section 4.4) used to describe the sensor(s) that captured the information used during the analysis.

Depending on the tools/devices used to detect incidents, an Analyzer may rely on the output from a single sensor or from multiple sensors to generate alerts. In addition, the Analyzer and Sensor may actually be part of the same physical device and may share some of their attributes (e.g. IP, Hostname, Model, etc.).

**Source**

Zero or more. Instances of the Source class (Section 4.5) used to describe the source(s) of the incident (e.g. attackers, faulty device, etc.).

**Target**

Zero or more. Instances of the Target class (Section 4.6) used to describe the target(s) of the incident, i.e. the impacted devices/users/services.

**Vector**

Zero or more. Instances of the Vector class (Section 4.7) used to describe the means which were employed by the sources to disrupt the targets.

E.g. to describe a car crashing into a building and resulting in service loss.

**Observable**

Zero or more. Instances of the Observable class (Section 4.9) used to describe a feature or phenomenon that can be observed or measured for the purposes of detecting malicious behavior.

This may include anything that may help security analysts in their understanding and analysis of the incident.

If the information is available as an electronic file, the Attachment class (Section 4.8) SHOULD be used instead.

**Attachment**

Zero or more. Instances of the Attachment class (Section 4.8) used to describe the electronic artifacts captured in relation with the incident.

The intent of the Attachment class is to keep track of the electronic files left as a trail during the incident. This may include things like on-disk files (e.g. malware samples), network packet captures, videos or still images from a camera feed, etc.

If the information is not readily-available as an electronic file, consider using the Observable class (Section 4.9) instead.

The Alert class has the following attributes:

**Version**

Mandatory. The version of the IDMEF format in use by this alert.

For this version of the IDMEF specification, this is the constant string "2.0".

**ID**

Mandatory. Unique identifier for the alert.

**Entity**

Optional. Tenant ID to support multi-tenancy (e.g. decentralized infrastructure, local agency, subsidiary company, etc.).

Should be used when there are multiple sites/locations or multiple tenants (e.g. by Managed Security Services Providers).

**Category**

Optional. The incident's category & subcategory as listed in [ENISA-RIST] using the format "category.subcategory" (e.g. "Attempt.Exploit").

| Rank | Keyword | Description |
|---|---|---|
| 0 | Abusive.Spam | Or 'Unsolicited Bulk Email', this means that the recipient has not granted verifiable permission for the message to be sent and that the message is sent as part of a larger collection of messages, all having a functionally comparable content. This IOC refers to resources, which make up a SPAM infrastructure, be it a harvesters like address verification, URLs in spam e-mails etc. |
| 1 | Abusive.Harassment | Discretization or discrimination of somebody, e.g. cyber stalking, racism or threats against one or more individuals. |
| 2 | Abusive.Illicit | Child Sexual Exploitation (CSE), Sexual content, glorification of violence, etc. |
| 3 | Malicious.System | System infected with malware, e.g. PC, smartphone or server infected with a rootkit. Most often this refers to a connection to a sinkholed C2 server |

| Rank | Keyword | Description |
|---|---|---|
| 4 | Malicious.Botnet | Command-and-control server contacted by malware on infected systems. |
| 5 | Malicious.Distribution | URI used for malware distribution, e.g. a download URL included in fake invoice malware spam or exploit-kits (on websites). |
| 6 | Malicious.Configuration | URI hosting a malware configuration file, e.g. web-injects for a banking trojan. |
| 7 | Recon.Scanning | Attacks that send requests to a system to discover weaknesses. This also includes testing processes to gather information on hosts, services and accounts. Examples: fingerd, DNS querying, ICMP, SMTP (EXPN, RCPT, ...), port scanning. |
| 8 | Recon.Sniffing | Observing and recording of network traffic (wiretapping). |
| 9 | Recon.SocialEngineering | Gathering information from a human being in a non-technical way (e.g. lies, tricks, bribes, or threats). |
| 10 | Attempt.Exploit | An attempt to compromise a system or to disrupt any service by exploiting vulnerabilities with a standardised identifier such as CVE name (e.g. buffer overflow, backdoor, cross site scripting, etc.) |
| 11 | Attempt.Login | Multiple login attempts (Guessing / cracking of passwords, brute force). This IOC refers to a resource, which has been observed to perform brute-force attacks over a given application protocol. |
| 12 | Attempt.NewSignature | An attack using an unknown exploit. |
| 13 | Intrusion.AdminCompromise | |

| Rank | Keyword | Description |
|---|---|---|
| | | Compromise of a system where the attacker gained administrative privileges. |
| 14 | Intrusion.UserCompromise | Compromise of a system using an unprivileged (user/service) account. |
| 15 | Intrusion.AppCompromise | Compromise of an application by exploiting (un-)known software vulnerabilities, e.g. SQL injection. |
| 16 | Intrusion.SysCompromise | Compromise of a system, e.g. unauthorised logins or commands. This includes compromising attempts on honeypot systems. |
| 17 | Intrusion.Burglary | Physical intrusion, e.g. into corporate building or data-centre. |
| 18 | Availability.DoS | Denial of Service attack, e.g. sending specially crafted requests to a web application which causes the application to crash or slow down. |
| 19 | Availability.DDoS | Distributed Denial of Service attack, e.g. SYN-Flood or UDP-based reflection/amplification attacks. |
| 20 | Availability.Misconf | Software misconfiguration resulting in service availability issues, e.g. DNS server with outdated DNSSEC Root Zone KSK. |
| 21 | Availability.Theft | Physical theft, e.g. stolen laptop computer, stolen USB key, stolen paper document, etc. |
| 22 | Availability.Sabotage | Physical sabotage, e.g cutting wires or malicious arson. |
| 23 | Availability.Outage | Outage caused e.g. by air condition failure or natural disaster. |
| 24 | Availability.Failure | Failure, malfunction (e.g. : bug, wear, faults, etc.) |
| 25 | Information. UnauthorizedAccess | Unauthorised access to information, e.g. by abusing stolen login credentials for a |

| Rank | Keyword | Description |
|---|---|---|
| | | system or application, intercepting traffic or gaining access to physical documents. |
| 26 | Information. UnauthorizedModification | Unauthorised modification of information, e.g. by an attacker abusing stolen login credentials for a system or application or a ransomware encrypting data. Also includes defacements. |
| 27 | Information.DataLoss | Loss of data, e.g. caused by harddisk failure or physical theft. |
| 28 | Information.DataLeak | Leaked confidential information like credentials or personal data. |
| 29 | Fraud.UnauthorizedUsage | Using resources for unauthorised purposes including profit-making ventures, e.g. the use of e-mail to participate in illegal profit chain letters or pyramid schemes. |
| 30 | Fraud.Copyright | Offering or Installing copies of unlicensed commercial software or other copyright protected materials (Warez). |
| 31 | Fraud.Masquerade | Type of attack in which one entity illegitimately impersonates the identity of another in order to benefit from it. |
| 32 | Fraud.Phishing | Masquerading as another entity in order to persuade the user to reveal private credentials. This IOC most often refers to a URL, which is used to phish user credentials. |
| 33 | Vulnerable.Crypto | Publicly accessible services offering weak crypto, e.g. web servers susceptible to POODLE/FREAK attacks. |
| 34 | Vulnerable.DDoS | Publicly accessible services that can be abused for conducting DDoS reflection/ |

| Rank | Keyword | Description |
|---|---|---|
| | | amplification attacks, e.g. DNS open-resolvers or NTP servers with monlist enabled. |
| 35 | Vulnerable.Surface | Potentially unwanted publicly accessible services, e.g. Telnet, RDP or VNC. |
| 36 | Vulnerable.Disclosure | Publicly accessible services potentially disclosing sensitive information, e.g. SNMP or Redis. |
| 37 | Vulnerable.System | A system which is vulnerable to certain attacks. Example: misconfigured client proxy settings (example: WPAD), outdated operating system version, XSS vulnerabilities, etc. |
| 38 | Geophysical.Earthquake | A hazard originating from solid earth. This term is used interchangeably with the term geological hazard. |
| 39 | Geophysical.MassMovement | A hazard originating from solid earth. This term is used interchangeably with the term geological hazard. |
| 40 | Geophysical.Volcanic | A hazard originating from solid earth. This term is used interchangeably with the term geological hazard. |
| 41 | Meteorological. Temperature | A hazard caused by short-lived, micro- to meso-scale extreme weather and atmospheric conditions that last from minutes to days. |
| 42 | Meteorological.Fog | A hazard caused by short-lived, micro- to meso-scale extreme weather and atmospheric conditions that last from minutes to days. |
| 43 | Meteorological.Storm | A hazard caused by short-lived, micro- to meso-scale extreme weather and atmospheric conditions that last from minutes to days. |
| 44 | Hydrological.Flood | A hazard caused by the occurrence, movement, and |

| Rank | Keyword | Description |
|---|---|---|
| | | distribution of surface and subsurface freshwater and saltwater. |
| 45 | Hydrological.Landslide | A hazard caused by the occurrence, movement, and distribution of surface and subsurface freshwater and saltwater. |
| 46 | Hydrological.Wave | A hazard caused by the occurrence, movement, and distribution of surface and subsurface freshwater and saltwater. |
| 47 | Climatological.Drought | A hazard caused by long-lived, meso- to macro-scale atmospheric processes ranging from intra-seasonal to multi-decadal climate variability. |
| 48 | Climatological. LakeOutburst | A hazard caused by long-lived, meso- to macro-scale atmospheric processes ranging from intra-seasonal to multi-decadal climate variability. |
| 49 | Climatological.Wildfire | A hazard caused by long-lived, meso- to macro-scale atmospheric processes ranging from intra-seasonal to multi-decadal climate variability. |
| 50 | Biological.Epidemic | A hazard caused by the exposure to living organisms and their toxic substances (e.g. venom, mold) or vector-borne diseases that they may carry. Examples are venomous wildlife and insects, poisonous plants, and mosquitoes carrying disease-causing agents such as parasites, bacteria, or viruses (e.g. malaria). |
| 51 | Biological.Insect | A hazard caused by the exposure to living organisms and their toxic substances (e.g. venom, mold) or vector-borne diseases that they may carry. Examples are venomous |

| Rank | Keyword | Description |
|---|---|---|
| | | wildlife and insects, poisonous plants, and mosquitoes carrying disease-causing agents such as parasites, bacteria, or viruses (e.g. malaria). |
| 52 | Biological.Animal | A hazard caused by the exposure to living organisms and their toxic substances (e.g. venom, mold) or vector-borne diseases that they may carry. Examples are venomous wildlife and insects, poisonous plants, and mosquitoes carrying disease-causing agents such as parasites, bacteria, or viruses (e.g. malaria). |
| 53 | Extraterrestrial.Impact | A hazard caused by asteroids, meteoroids, and comets as they pass near-earth, enter the Earth's atmosphere, and/or strike the Earth, and by changes in interplanetary conditions that effect the Earth's magnetosphere, ionosphere, and thermosphere. |
| 54 | Extraterrestrial. SpaceWeather | A hazard caused by asteroids, meteoroids, and comets as they pass near-earth, enter the Earth's atmosphere, and/or strike the Earth, and by changes in interplanetary conditions that effect the Earth's magnetosphere, ionosphere, and thermosphere. |
| 55 | Other.Uncategorised | All incidents which don't fit in one of the given categories should be put into this class or the incident is not categorised. |
| 56 | Other.Undetermined | The categorisation of the incident is unknown/ undetermined. |
| 57 | Test.Test | Meant for testing. |

Table 1: Incident taxonomy

**Cause**

Optional. Alert cause, if known at the time of detection.

If unknown, this key SHOULD NOT be defined by the analyzer and may be filled later on by a manager or a human operator.

| Rank | Keyword | Description |
|---|---|---|
| 0 | Normal | The event is related to an expected phenomenon or to a phenomenon that does not qualify as out of the ordinary. |
| 1 | Error | The event is related to a human error. |
| 2 | Malicious | The event is related to malicious code or malicious actions. |
| 3 | Malfunction | The event is related to a device or service malfunction. |
| 4 | Natural | The event is related to a natural phenomenon. |
| 5 | Unknown | The cause of the event is unknown. |

Table 2: Incident causes

**Description**

Optional. Short free text human-readable description.

**Status**

Optional. Alert state in the overall alert lifecycle.

| Rank | Keyword | Description |
|---|---|---|
| 0 | Event | |
| 1 | Incident | |

Table 3: Incident statuses

**Severity**

Optional. Severity of the alert.

| Rank | Keyword | Description |
|---|---|---|
| 0 | Unknown | |
| 1 | Info | |
| 2 | Low | |
| 3 | Medium | |
| 4 | High | |

Table 4: Incident severities

**Confidence**

Optional. A floating-point value between 0 and 1 indicating the analyzer's confidence in its own reliability of this particular detection, where 0 means that the detection is surely incorrect while 1 means there is no doubt about the detection made.

**Note**

Optional. Free text human-readable additional note, possibly a longer description of the incident if is not already obvious.

**CreateTime**

Mandatory. Timestamp indicating when the message was created. May point out delay between detection and processing of the events.

**StartTime**

Optional. Timestamp indicating the deduced start of the event.

In case the event is not part of a series, this attribute MAY instead be set to the timestamp initially present in the event (if any).

**CeaseTime**

Optional. Timestamp indicating the deduced end of the event.

**DeleteTime**

Optional. Timestamp indicating when the message must be deleted.

This attribute MUST be specified if the message has to be deleted after this date, e.g. for technical, organizational or ethical reasons.

**AltNames**

Optional. Alternative identifiers; strings which help pair the event to internal systems' information (for example ticket IDs inside a request tracking systems).

**AltCategory**

Optional. Alternate categories from a reference other than [ENISA-RIST] (e.g. MISP, MITRE ATT@CK or another proprietary/internal reference).

**Ref**

Optional. References to sources of information related to the alert and/or vulnerability, and specific to this alert.

This MAY be a URL to additional info, or a URN in a registered or unregistered ad-hoc namespace bearing reasonable information value and uniqueness, such as "urn:cve:CVE-2013-2266".

**CorrelID**

Optional. Identifiers for the messages which were used as information sources to create this message, in case the message has been created based on correlation/analysis/deduction from other messages.

**AggrCondition**

Optional. A list of IDMEF fields used to aggregate events. The values for these fields will be the same in all aggregated events.

This attribute should mostly be set by intermediary nodes, which detect duplicates, or aggregate events, spanning multiple detection windows, into a longer one.

The "StartTime" and "CeaseTime" attributes are used in conjunction with this attribute to describe the aggregation window.

**PredID**

Optional. A list containing the identifiers of previous messages which are obsoleted by this message.

The obsoleted alerts SHOULD NOT be used anymore. This field can be used to "update" an alert.

**RelID**

Optional. A list containing the identifiers of other messages related to this message.

### 4.3. The Analyzer Class

The Analyzer class describes the module that has analyzed the data captured by the sensors, identified an event of interest and decided to create an alert.

```
+-----------------------+
|       Analyzer        |
+-----------------------+
| IP       IP           |
| STRING   Name         |
| STRING   Hostname     |
| STRING   Model        |
| ENUM[]   Type         |
| ENUM[]   Category     |
| ENUM[]   Data         |
| ENUM[]   Method       |
| GEOLOC   GeoLocation  |
| UNLOCODE UnLocation   |
| STRING   Location     |
+-----------------------+
```

Figure 4: The Analyzer class

The Analyzer class has the following attributes:

**IP**

Mandatory. Analyzer IP address.

**Name**

Mandatory. Name of the analyzer, which must be reasonably unique, however still bear some meaningful sense.

This attribute usually denotes the hierarchy of organizational units the detector belongs to and its own name. It MAY also be used to distinguish multiple analyzers running with the same IP address.

**Hostname**

Optional. Hostname of this analyzer.

SHOULD be a fully-qualified domain name.

**Model**

Mandatory. Analyzer model description (usually its generic name, brand and version).

**Type**

Optional. Analyzer type.

| Rank | Keyword | Description |
|---|---|---|
| 0 | Cyber | The analyzer specializes in the detection of cyber incidents |
| 1 | Physical | The analyzer specializes in the detection of physical incidents |
| 2 | Availability | The analyzer specializes in the detection of availability incidents |
| 3 | Combined | The analyzer specilizes in detections that combine data from multiple domains (e.g. a combination of Cyber and Availability data) |

Table 5: Analyzer types

**Category**

Mandatory. Analyzer categories.

| Rank | Keyword | Description |
|---|---|---|
| 0 | 1DLiS | 1D LIDAR Sensor |
| 1 | 2DLiS | 2D LIDAR Sensor |
| 2 | 3DLiS | 3D LIDAR Sensor |

| Rank | Keyword | Description |
|---|---|---|
| 3 | 1DLaS | 1D Laser Sensor |
| 4 | 2DLaS | 2D Laser Sensor |
| 5 | 3DLaS | 3D Laser Sensor |
| 6 | VAD | Voice Activity Detection |
| 7 | HAR | Human Activity Detection |
| 8 | FRC | Face Recognition Camera |
| 9 | VNIR | Visible and Near-InfraRed |
| 10 | SWIR | Short Wavelength InfraRed |
| 11 | MWIR | Middle Wavelength InfraRed |
| 12 | LWIR | Long Wavelength InfraRed |
| 13 | ADS | Anti-Drone System |
| 14 | ODC | Object Detection Camera |
| 15 | DDOS | Anti-DDoS (Distributed Denial of Service) protection |
| 16 | SPAM | Spam detection, phishing detection, etc. |
| 17 | AV | Signature-based virus/malware detection |
| 18 | EDR | Endpoint Detection and Response |
| 19 | FW | Firewall |
| 20 | NIDS | Network Intrusion Detection System |
| 21 | HIDS | Host Intrusion Detection System |
| 22 | WIDS | Wi-Fi Intrusion Detection System |
| 23 | PROX | Proxy, e.g. detection of violations to the company's security policy |
| 24 | WAF | Web Application Firewall |
| 25 | HPT | Honeypot |
| 26 | LOG | Log analyzer |
| 27 | IAM | Identity and Access Management tool |
| 28 | VPN | Devices/tools related to Virtual Private Network |
| 29 | ETL | Extract-Transform-Load tools |
| 30 | RASP | Runtime Application Self-Protection |
| 31 | BAST | Clientless Remote Desktop Gateway / administration bastions |
| 32 | NAC | Devices/tools related to Network Access Control |
| 33 | SIEM | Security Information and Event Management systems |
| 34 | NMS | Network Management Systems |

Table 6: Analyzer categories

**Data**

Mandatory. Type of data analyzed during the detection.

| Rank | Keyword | Description |
|---|---|---|
| 0 | Light | |
| 1 | Noise | |
| 2 | Touch | |

| Rank | Keyword | Description |
|---|---|---|
| 3 | Images | |
| 4 | Vibrations | |
| 5 | Lidar | |
| 6 | Thermic | |
| 7 | Seismic | |
| 8 | Temperature | |
| 9 | Rain | |
| 10 | Water | |
| 11 | Humidity | |
| 12 | Particles | |
| 13 | Contact | |
| 14 | MagneticField | |
| 15 | Acoustics | |
| 16 | Fog | |
| 17 | External | |
| 18 | Reporting | |
| 19 | Connection | |
| 20 | Datagram | |
| 21 | Content | |
| 22 | Data | |
| 23 | File | |
| 24 | Flow | |
| 25 | Log | |
| 26 | Protocol | |
| 27 | Host | |
| 28 | Network | |
| 29 | Alert | |
| 30 | Relay | |
| 31 | Auth | |
| 32 | SNMP | |

Table 7: Analyzer data

**Method**

Mandatory. Detection method.

| Rank | Keyword | Description |
|---|---|---|
| 0 | Biometric | |
| 1 | Policy | |
| 2 | Heat | |
| 3 | Movement | |
| 4 | Blackhole | |
| 5 | Signature | |
| 6 | Statistical | |
| 7 | Heuristic | |
| 8 | Integrity | |
| 9 | Honeypot | |

| Rank | Keyword | Description |
|---|---|---|
| 10 | Tarpit | |
| 11 | Recon | |
| 12 | Correlation | |
| 13 | Monitor | |
| 14 | AI | |
| 15 | Threshold | |

Table 8: Analyzer methods

**GeoLocation**
Optional. GPS coordinates for the analyzer.

**UnLocation**
Optional. Standard UN/Locode for the analyzer.

**Location**
Optional. Internal name for the location of the analyzer.

## 4.4. The Sensor Class

The Sensor class describes the module that captured the data before sending it to an analyzer. The Sensor may be a subpart of the Analyzer.

```
+----------------------+
|        Sensor        |
+----------------------+
| IP       IP          |
| STRING   Name        |
| STRING   Hostname    |
| STRING   Model       |
| UNLOCODE UnLocation  |
| STRING   Location    |
| STRING   CaptureZone |
+----------------------+
```

Figure 5: The Sensor class

The Sensor class has the following attributes:

**IP**
Mandatory. The sensor's IP address.

**Name**
Mandatory. Name of the sensor, which must be reasonably unique, however still bear some meaningful sense.

This attribute usually denotes the hierarchy of organizational units the sensor belongs to and its own name. It MAY also be used to distinguish multiple sensors running with the same IP address.

**Hostname**

Optional. The sensor's hostname.

This SHOULD be a fully qualified domain name, but may not conform exactly because values extracted from logs, messages, DNS, etc. may themselves be malformed.

An empty string MAY be used to explicitly state that this value was inquired but not found (missing DNS entry).

**Model**

Mandatory. The sensor model's description (usually its generic name, brand and version).

**UnLocation**

Optional. Standard UN/Locode for the sensor.

**Location**

Optional. Internal name for the location of the sensor.

**CaptureZone**

Optional. A string that describes the "capture zone" of the sensor, as a JSON-serialized string.

Depending on the type of sensor, the capture zone may for instance refer to:

- A JSON object describing a camera's settings (elevation, horizontal and vertical field of view, azimuth, etc.)

- A description of the IP network where packet capture is taking place.

### 4.5. The Source Class

The Source class describes the source(s) of the event(s) leading up to the alert.

In this context, the Source always refers to the attacker, which may be different from the source in the context of a network connection. For instance, when a user connects to a webserver spreading malwares, the webserver will be listed as the IDMEF Source, even though it was initially the destination of the underlying HTTP(S) connection.

```
+------------------------+
|         Source         |
+------------------------+
| IP          IP         |
| STRING      Hostname   |
| STRING      Note       |
| STRING[]    TI         |
| STRING      User       |
| EMAIL       Email      |
| PROTOCOL[] Protocol    |
| INT[]       Port       |
| GEOLOC      GeoLocation |
| UNLOCODE    UnLocation |
| STRING      Location   |
| ID[]        Attachment |
| ID[]        Observable |
+------------------------+
```

Figure 6: The Source class

The Source class has the following attributes:

**IP**

Optional. Source IP address.

**Hostname**

Optional. Hostname of this source.

This SHOULD be a fully qualified domain name, but may not conform exactly because values extracted from logs, messages, DNS, etc. may themselves be malformed.

An empty string MAY be used to explicitly state that this value was inquired but not found (missing DNS entry).

**Note**

Optional. Free text human-readable additional note for this source.

**TI**

Optional. Threat Intelligence data about the source.

Values in this list MUST use the format "attribute:origin", where "attribute" refers to the attribute inside this source found inside a Threat Intelligence database, and "origin" contains a short identifier for the Threat Intelligence database. E.g. "IP:Dshield".

Please note that the same attribute may appear multiple times inside the list (because a match was found in multiple Threat Intelligence databases).

**User**

Optional. User ID or login responsible for the alert.

**Email**

Optional. Email address responsible for the alert.

E.g. the value of the "Reply-To" or "From" header inside a phishing e-mail.

**Protocol**

Optional. Protocols related to connections from/to this source.

If several protocols are stacked, they MUST be ordered from the lowest (the closest to the medium) to the highest (the closest to the application) according to the ISO/OSI model.

**Port**

Optional. Source ports involved in the alert.

Values in this list MUST be integers and MUST be in the range 1-65535.

**GeoLocation**

Optional. GPS coordinates for the source.

**UnLocation**

Optional. Standard UN/Locode for the source.

**Location**

Optional. Internal name for the location of the source.

**Attachment**

Optional. Identifiers for attachments related to this source.

Each identifier listed here MUST match the "Name" attribute for one of the attachments described using the Attachment class (Section 4.8).

**Observable**

Optional. Identifiers for observables related to this source.

Each identifier listed here MUST match the "Name" attribute for one of the observables described using the Observable class (Section 4.9).

### 4.6. The Target Class

The Target class describes the target(s) of the event(s) leading up to the alert.

In this context, the Target always refers to the potential victim, which may be different from the destination in the context of a network connection. For instance, when a user connects to a webserver spreading malwares, the user will be listed as the IDMEF Target, even though it was initially the source of the underlying HTTP(S) connection.

```
+------------------------+
|         Target         |
+------------------------+
| IP          IP         |
| STRING      Hostname   |
| STRING      Note       |
| STRING      Service    |
| STRING      User       |
| EMAIL       Email      |
| INT[]       Port       |
| GEOLOC      GeoLocation |
| UNLOCODE    UnLocation |
| STRING      Location   |
| ID[]        Attachment |
| ID[]        Observable |
+------------------------+
```

Figure 7: The Target class

The Target class has the following attributes:

**IP**
Optional. Target IP address.

**Hostname**
Optional. Hostname of this target.

This SHOULD be a fully qualified domain name, but may not conform exactly because values extracted from logs, messages, DNS, etc. may themselves be malformed.

An empty string MAY be used to explicitly state that this value was inquired but not found (missing DNS entry).

**Note**
Optional. Free text human-readable additional note for this target.

**Service**

Optional. Service or process impacted by the alert.

**User**

Optional. User ID or login targeted by the alert.

**Email**

Optional. Email address targeted by the alert.

E.g. the value of the "To" header inside a phishing e-mail.

**Port**

Optional. Target ports involved in the alert.

Values in this list MUST be integers and MUST be in the range 1-65535.

**GeoLocation**

Optional. GPS coordinates for the target.

**UnLocation**

Optional. Standard UN/Locode for the target.

**Location**

Optional. Internal name for the location of the target.

**Attachment**

Optional. Identifiers for attachments related to this target.

Each identifier listed here MUST match the "Name" attribute for one of the attachments described using the Attachment class (Section 4.8).

**Observable**

Optional. Identifiers for observables related to this target.

Each identifier listed here MUST match the "Name" attribute for one of the observables described using the Observable class (Section 4.9).

### 4.7. The Vector Class

The Vector class describes the vector(s) of the event(s) leading up to the alert. o Name, location, description, ...

```
+-------------------------+
|          Vector         |
+-------------------------+
| ENUM[]     Category     |
| STRING     Name         |
| ENUM       Size         |
| STRING     Note         |
| STRING[]   TI           |
| GEOLOC     GeoLocation  |
| FLOAT      GeoRadius    |
| UNLOCODE   UnLocation   |
| STRING     Location     |
| ID[]       Attachment   |
| ID[]       Observable   |
+-------------------------+
```

Figure 8: The Vector class

The Vector class has the following attributes:

**Category**

Mandatory. Category for the detected "vector".

FIXME: Les valeurs du domaine cyber n'ont pas ete ajoutees car elles semblent redondantes avec la notion d'Observable.

| Rank | Keyword | Description |
|---|---|---|
| 0 | Unknown | |
| 1 | Face | |
| 2 | RunningMan | |
| 3 | Human | |
| 4 | Man | |
| 5 | Woman | |
| 6 | Children | |
| 7 | Animal | |
| 8 | Object | |
| 9 | Blast | |
| 10 | Fire | |
| 11 | Wind | |
| 12 | Snow | |
| 13 | Rain | |
| 14 | Chemical | |
| 15 | Smoke | |
| 16 | Vapors | |
| 17 | Drug | |
| 18 | Device | |
| 19 | Drone | |
| 20 | Car | |

| Rank | Keyword | Description |
|---|---|---|
| 21 | Truck | |
| 22 | Vehicle | |
| 23 | Bird | |
| 24 | Storm | |
| 25 | HighTemperature | |
| 26 | Artifact | |
| 27 | Autonomous System | |
| 28 | Directory | |
| 29 | Domain Name | |
| 30 | Email Address | |
| 31 | Email Message | |
| 32 | File | |
| 33 | IPv4 Address | |
| 34 | IPv6 Address | |
| 35 | Mutex | |
| 36 | Network Traffic | |
| 37 | Process | |
| 38 | URL | |
| 39 | User Account | |
| 40 | Windows Registry Key | |
| 41 | X509 Certificate | |

Table 9: Vector categories

**Name**

Optional. Name of the detected vector or "Unknown".

Please note that this name does not need to be unique across vectors.

**Size**

Optional. Rough estimate of the detected vector's size.

| Rank | Keyword | Description |
|---|---|---|
| 0 | Small | For things like a dog, a small drone, etc. |
| 1 | Medium | For things like a person |
| 2 | Large | For things like a car, a truck, etc. |
| 3 | Huge | For things like a big crowd, a storm, etc. |

Table 10: Vector sizes

**Note**

Optional. Free text human-readable additional note for this vector.

**TI**

Optional. Threat Intelligence data about the vector.

Values in this list MUST use the format "attribute:origin", where "attribute" refers to the attribute inside this vector found inside a Threat Intelligence database, and "origin" contains a short identifier for the Threat Intelligence database. E.g. "Name:FBI-Wanted".

Please note that the same attribute may appear multiple times inside the list (because a match was found in multiple Threat Intelligence databases).

**GeoLocation**

Optional. GPS coordinates for the vector.

**GeoRadius**

Optional. Estimated radius around the provided geolocation in meters.

This attribute can be interpreted as an error margin related to the detection of this vector.

**UnLocation**

Optional. Standard UN/Locode for the vector.

**Location**

Optional. Internal name for the location of the vector.

**Attachment**

Optional. Identifiers for attachments related to this vector.

Each identifier listed here MUST match the "Name" attribute for one of the attachments described using the Attachment class (Section 4.8).

**Observable**

Optional. Identifiers for observables related to this vector.

Each identifier listed here MUST match the "Name" attribute for one of the observables described using the Observable class (Section 4.9).

### 4.8. The Attachment Class

The Attachment class contains additional data which was captured in relation with the event.

```
+-----------------------------+
|          Attachment         |
+-----------------------------+
| ID          Name            |
| STRING      FileName        |
| HASH[]      Hash            |
| INT         Size            |
| URI[]       Ref             |
| URI[]       ExternalURI     |
| STRING      Note            |
| MEDIATYPE   ContentType     |
| STRING      ContentEncoding |
| STRING      Content         |
+-----------------------------+
```

Figure 9: The Attachment class

The Attachment class has the following attributes:

**Name**

Mandatory. A unique identifier among attachments that can be used to reference this attachment from other classes using the "Attachment" attribute.

**FileName**

Optional. Attachment filename.

This will usually be the original name of the captured file or the name of the file containing the captured content (e.g. a packet capture file).

**Hash**

Optional. A list of hash results for the attachment's Content.

The values in this list are computed by taking the raw value of the attachment's "Content" attribute. The hash result is computed before any other transformation (e.g. Base64 encoding) is applied to the content, so that a receiving IDMEF system may reverse the transformation, apply the same hashing function and obtain the same hash result. See also the definition for the "ContentEncoding" attribute below.

It is RECOMMENDED that compatible implementations use one of the hashing functions from the SHA-2 [RFC6234] or SHA-3 [NIST.FIPS.202] families to compute the hash results in this list.

**Size**

Optional. Length of the content (in bytes).

This value MUST be a non-negative integer.

**Ref**

Optional. References to sources of information related to the alert and/or vulnerability, and specific to this attachment.

**ExternalURI**

Optional. If the attachment's content is available and/or recognizable from an external resource, this is the URI (usually a URL) to that resource.

This MAY also be a URN in a registered or unregistered ad-hoc namespace bearing reasonable information value and uniqueness, such as "urn:mhr:55eaf7effadc07f866d1eaed9c64e7ee49fe081a" or "magnet:?xt=urn:sha1:YNCKHTQCWBTRNJIV4WNAE52SJUQCZO5C".

**Note**

Optional. Free text human-readable additional note for this attachment.

**ContentType**

Optional. Internet Media Type of the attachment.

For compatibility reasons, implementations SHOULD prefer one of the well-known media types registered in IANA .

**ContentEncoding**

Optional. Content encoding.

The following encodings are defined in this version of the specification:

- "json": The content refers to a JSON object which has been serialized to a string using the serialization procedure defined in [RFC8259].

- "base64": The content has been serialized using the Base64 encoding defined in [RFC4648].

The "base64" encoding SHOULD be used when the content contains binary data. If omitted, the "json" encoding MUST be assumed.

**Content**

Optional. The attachment's content, in case it is directly embedded inside the message.

For large attachments, it is RECOMMENDED that implementations make use of the "ExternalURI" attribute to refererence a copy of the content saved in an external storage mechanism.

### 4.9. The Observable Class

The Observable class describes a feature or phenomenon that can be observed or measured for the purposes of detecting malicious behavior.

```
+------------------+
|    Observable    |
+------------------+
| ID     Name      |
| STRING Reference |
| STRING Content   |
+------------------+
```

Figure 10: The Observable class

The Observable class has the following attributes:

**Name**

Mandatory. A unique identifier among observables that can be used to reference this observable from other classes using the "Observable" attribute.

**Reference**

Optional. Name of the reference where the observable is specified.

This attribute is meant to help implementations in identifying supported observables.

**Content**

Mandatory. Observable content.

### 4.10. The JavaScript Object Notation Serialization Method

This serialization method aims to convert IDMEFv2 messages to a format that is easy to parse and process, both by software/hardware processors, as well as humans. It relies on the the JavaScript Object Notation (JSON) Data Interchange Format defined in [RFC8259].

Conforming implementations MUST implement all the requirements specified in [RFC8259].

In addition, the following rules MUST be observed when serializing an IDMEFv2 message:

- The top-level Alert class (Section 4.2) is represented as a JSON object ([RFC8259]). This JSON object is returned to the calling process at the end of the serialization process.

- Aggregate classes are represented as JSON objects and stored as members of the top-level JSON object, using the same name as in the IDMEF data model. E.g. the appears under the name "Analyzer" inside the top-level JSON object.

- Attributes are stored as members of the JSON object representing the class they belong to, using the same name as in the IDMEF data model. E.g. the "Version" attribute from the is stored under the name "Version" inside the top-level JSON object.

- Lists from the IDMEF data model are represented as JSON arrays ([RFC8259]). This also applies to aggregate classes where a list is expected. E.g. the "Sensor" member inside the top-level JSON object contains a list of objects, where each object represents an instance of the .

- The various string-based data types listed in Section 3 are represented as JSON strings ([RFC8259]). Please note that the issues outlined in [RFC8259] regarding strings processing also apply here.

- IDMEF attributes with the "NUMBER" data type are represented as JSON numbers ([RFC8259]).

### 4.11. Attributes completeness

The next table shows when each attributes is required depending on it's Type: physical, cyber or availability.

Legend:

- R: REQUIRED

- r: Recommanded

- o: Optional

- NA: Not Applicable

| **Attributes** | **Type** | **Phy** | **Cyb** | **Avail** |
|---|---|---|---|---|
| **Alert** | | | | |
| Version | String | R | R | R |
| ID | UUID | R | R | R |
| Entity | String | o | o | o |
| Category | Array of ENUM | r | r | r |
| Cause | ENUM | r | r | r |
| Description | String | r | r | r |
| Status | ENUM | r | r | r |

| **Attributes** | **Type** | **Phy** | **Cyb** | **Avail** |
|---|---|---|---|---|
| **Alert** | | | | |
| Severity | ENUM | r | r | r |
| Confidence | Number | o | o | o |
| Note | String | o | o | o |
| CreateTime | Timestamp | R | R | R |
| StartTime | Timestamp | r | r | r |
| CeaseTime | Timestamp | o | o | o |
| DeleteTime | Timestamp | o | o | o |
| AltNames | Array of String | o | o | o |
| AltCategory | Array of String | o | o | o |
| Ref | Array of URI | o | o | o |
| CorrelID | Array of UUID | o | o | o |
| AggrCondition | Array of String | o | o | o |
| PredID | Array of UUID | o | o | o |
| RelID | Array of UUID | o | o | o |

Table 11: Attributes completness - Alert

| **Attributes** | **Type** | **Phy** | **Cyb** | **Avail** |
|---|---|---|---|---|
| **Analyzer** | **Class** | **R** | **R** | **R** |
| IP | IPAddress | R | R | R |
| Name | String | R | R | R |
| Hostname | String | r | r | r |
| Type | ENUM | r | r | r |
| Model | String | R | R | R |
| Category | Array of ENUM | R | R | R |
| Data | Array of ENUM | R | R | R |
| Method | Array of ENUM | R | R | R |
| GeoLocation | GeoLocation | r | o | o |
| UnLocation | UN/LOCODE | o | o | o |
| Location | String | o | o | o |

Table 12: Attributes completness - Analyzer

| **Attributes** | **Type** | **Phy** | **Cyb** | **Avail** |
|---|---|---|---|---|
| **Sensor** | **Array of Class** | **o** | **o** | **o** |
| IP | IPAddress | R | R | R |
| Name | String | R | R | R |
| Hostname | String | r | r | r |
| Model | String | R | R | R |
| UnLocation | UN/LOCODE | o | o | o |
| Location | String | o | o | o |
| CaptureZone | String | o | o | o |

Table 13: Attributes completness - Sensor

| **Attributes** | **Type** | **Phy** | **Cyb** | **Avail** |
|---|---|---|---|---|
| **Source** | **Array of Class** | **o** | **o** | **o** |
| UnLocation | UN/LOCODE | o | o | NA |

| **Attributes** | **Type** | **Phy** | **Cyb** | **Avail** |
|---|---|---|---|---|
| **Source** | **Array of Class** | **o** | **o** | **o** |
| Location | String | o | o | NA |
| GeoLocation | GeoLocation | NA | o | NA |
| Note | String | o | o | o |
| TI | Array of String | o | o | o |
| IP | IPAddress | NA | r | NA |
| Hostname | String | NA | r | NA |
| User | String | NA | o | NA |
| Email | String | NA | o | NA |
| Protocol | Array of ProtocolName | NA | o | NA |
| Port | Array of Port | NA | o | NA |
| Attachment | Array of AttachmentName | NA | o | NA |
| Observable | Array of ObservableName | NA | o | o |

Table 14: Attributes completness - Source

| **Attributes** | **Type** | **Phy** | **Cyb** | **Avail** |
|---|---|---|---|---|
| **Target** | **Array of Class** | **o** | **R** | **R** |
| UnLocation | UN/LOCODE | o | o | o |
| Location | String | r | o | o |
| GeoLocation | GeoLocation | o | o | o |
| Note | String | o | o | o |
| IP | IPAddress | o | r | R |
| Hostname | String | o | r | r |
| Service | String | NA | o | r |
| User | String | NA | o | NA |
| Email | String | NA | o | NA |
| Port | Array of Port | NA | o | o |
| Attachment | Array of AttachmentName | NA | o | o |
| Observable | Array of ObservableName | NA | o | o |

Table 15: Attributes completness - Target

| **Attributes** | **Type** | **Phy** | **Cyb** | **Avail** |
|---|---|---|---|---|
| **Vector** | **Array of Class** | **o** | **o** | **o** |
| Category | Array of ENUM | R | R | NA |
| TI | Array of String | o | o | NA |
| Name | String | o | NA | NA |
| Size | ENUM | o | NA | NA |
| UnLocation | UN/LOCODE | o | NA | NA |
| GeoLocation | GeoLocation | o | NA | NA |
| GeoRadius | Number | o | NA | NA |
| Location | String | r | NA | NA |
| Note | String | o | NA | NA |
| Attachment | Array of AttachmentName | o | o | o |
| Observable | Array of ObservableName | o | o | NA |

Table 16: Attributes completness - Vector

| **Attributes** | **Type** | **Phy** | **Cyb** | **Avail** |
|---|---|---|---|---|
| **Attachment** | **Array of Class** | **o** | **o** | **o** |
| Name | String | R | R | R |
| FileName | String | o | o | o |
| Hash | Array of Hashes | r | r | r |
| Size | Number | r | r | r |
| Ref | Array of URI | o | o | o |
| ExternalURI | Array of URI | o | o | o |
| Note | String | o | o | o |
| ContentType | MediaType | o | o | o |
| ContentEncoding | String | r | r | r |
| Content | String | o | o | o |

Table 17: Attributes completness - Attachment

| **Attributes** | **Type** | **Phy** | **Cyb** | **Avail** |
|---|---|---|---|---|
| **Observable** | **Array of Class** | **o** | **o** | **o** |
| Name | String | R | R | R |
| Reference | String | r | r | r |
| Content | String | R | R | R |

Table 18: Attributes completness - Observable

## 5. Security Considerations

This document describes a data representation for exchanging security-related information between incident detection system implementations. Although there are no security concerns directly applicable to the format of this data, the data itself may contain security-sensitive information whose confidentiality, integrity, and/or availability may need to be protected.

This suggests that the systems used to collect, transmit, process, and store this data should be protected against unauthorized use and that the data itself should be protected against unauthorized access.

The underlying messaging format and protocol used to exchange instances of the IDMEF MUST provide appropriate guarantees of confidentiality, integrity, and authenticity. The use of a standardized security protocol is encouraged.

The draft-poirotte-idmefv2-00.txt document defines the transportation of IDMEF over HTTPs that provides such security.

## 6. IANA Considerations

This document creates 10 identically structured registries to be managed by IANA:

- *Name of the parent registry: "Incident Detection Message Exchange Format v2 (IDMEF)"

- *URL of the registry: <http://www.iana.org/assignments/idmef2>

- *Namespace format: A registry entry consists of:

  - -Value. A value for a given IDMEF attribute. It MUST conform to the formatting specified by the IDMEF "ENUM" data type (Section 3.3.1).

  - -Description. A short description of the enumerated value.

  - -Reference. An optional list of URIs to further describe the value.

- *Allocation policy: Expert Review per [RFC8126]. This reviewer will ensure that the requested registry entry conforms to the prescribed formatting. The reviewer will also ensure that the entry is an appropriate value for the attribute per the information model (Section 4).

The registries to be created are named in the "Registry Name" column of Table 19. Each registry is initially populated with values and descriptions that come from an attribute specified in the IDMEF model (Section 4). The initial values for the Value and Description fields of a given registry are listed in "Initial Values". The "Initial Values" column points to a table in this document that lists and describes each enumerated value. Each enumerated value in the table gets a corresponding entry in a given registry. The initial value of the Reference field of every registry entry described below should be this document.

| Registry Name | Initial Values |
|---|---|
| Alert-Category | Table 1 (Alert class (Section 4.2)) |
| Alert-Cause | Table 2 (Alert class (Section 4.2)) |
| Alert-Severity | Table 4 (Alert class (Section 4.2)) |
| Alert-Status | Table 3 (Alert class (Section 4.2)) |
| Analyzer-Category | Table 6 (Alert class (Section 4.2)) |
| Analyzer-Data | Table 7 (Analyzer class (Section 4.3)) |
| Analayzer-Method | Table 8 (Analyzer class (Section 4.3)) |
| Analyzer-Type | Table 5 (Analyzer class (Section 4.3)) |
| Vector-Category | Table 9 (Vector (Section 4.7)) |

| Registry Name | Initial Values |
|---|---|
| Vector-Size | Table 10 (Vector (Section 4.7)) |

Table 19: IANA Enumerated Value Registries

## 7. Acknowledgement

Thanks to the core participants of the SECEF (SECurity Exchange Format) project :

*Herve Debar, Telecom SudParis

*Guillaume Hiet, CentraleSupelec

*Francois Dechelle, Teclib'

Thanks to the H2020 7SHIELD project (Safety and Security Standards of Space Systems, ground Segments and Satellite data assets, via prevention, detection, response and mitigation of physical and cyber threats) who implemented in real scale first versions of IDMEFv2 on different critical infrastructures around Europa helping greatly to improve it.

Thanks to the CESNET team for their work on the [IDEA0] format (based on IDMEFv1) which inspired multiples concepts to IDMEFv2. (<https://idea.cesnet.cz/en/index>)

## 8. References

### 8.1. Normative References

**[RFC5321]** Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <https://www.rfc-editor.org/info/rfc5321>.

**[RFC2046]** Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, DOI 10.17487/RFC2046, November 1996, <https://www.rfc-editor.org/info/rfc2046>.

**[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>.

**[RFC5322]** Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <https://www.rfc-editor.org/info/rfc5322>.

**[RFC3339]** Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <https://www.rfc-editor.org/info/rfc3339>.

**[RFC3986]** Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

**[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>.

**[RFC4291]** Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

**[RFC4648]** Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

**[RFC5234]** Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

**[RFC5952]** Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <https://www.rfc-editor.org/info/rfc5952>.

**[RFC8259]** Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/

RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

**[UNICODE]** Unicode Consortium, "Unicode Standard", version 14.0.0, 14 September 2021, <https://www.unicode.org/versions/Unicode14.0.0/>.

**[ENISA-RIST]** ENISA, "Reference Incident Classification Taxonomy", 26 January 2018, <https://www.enisa.europa.eu/publications/reference-incident-classification-taxonomy>.

**[IANA_media_types]** IANA, "Media Types", 10 October 2022, <http://www.iana.org/assignments/media-types>.

**[IANA_hash_function_text_names]** IANA, "Hash Function Textual Names", 21 April 2006, <http://www.iana.org/assignments/hash-function-text-names>.

**[UN-LOCODE]** UNECE, "UN/LOCODE Code List by Country and Territory", 6 July 2021, <https://unece.org/trade/cefact/unlocode-code-list-country-and-territory>.

## 8.2. Informative References

**[RFC4765]** Debar, H., Curry, D., and B. Feinstein, "The Intrusion Detection Message Exchange Format (IDMEF)", RFC 4765, DOI 10.17487/RFC4765, March 2007, <https://www.rfc-editor.org/info/rfc4765>.

**[RFC8126]** Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

**[RFC6234]** Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

**[NIST.FIPS.202]** Dworkin, Morris J., "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions", NIST NIST FIPS 202, DOI 10.6028/NIST.FIPS.202, July 2015, <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf>.

**[WGS84]** National Imagery and Mapping Agency, "Department of Defense World Geodetic System 1984: Its Definition and Relationships with Local Geodetic Systems", Third Edition, 1984, <https://apps.dtic.mil/sti/pdfs/ADA280358.pdf>.

**[IDEA0]**
CESNET, "Intrusion Detection Extensible Alert version 0", 25 September 2015, <https://idea.cesnet.cz/en/definition>.

## Appendix A. Examples

This section contains several examples of events/incidents which may be described using the IDMEF Data Model defined in.

For each example, the serialization method listed in Section 5 was used on the original IDMEF message to produce a JSON representation.

### A.1. Physical intrusion

Listing 1 describes an incident where an unidentified man was detected on company premises near the building where server room A is located.

```
{
  "Version": "2.0",
  "ID": "819df7bc-35ef-40d8-bbee-1901117370b1",
  "Description": "Potential intruder detected",
  "Severity": "Low",
  "Status": "Incident",
  "Cause": "Malicious",
  "CreateTime": "2021-05-10T16:52:13.075994+00:00",
  "StartTime": "2021-05-10T16:52:13+00:00",
  "Category": [
    "Intrusion.Burglary"
  ],
  "Analyzer": {
    "Name": "BigBrother",
    "Hostname": "bb.example.com",
    "Type": "Physical",
    "Model": "Big Brother v42",
    "Category": [
      "HAR",
      "FRC"
    ],
    "Data": [
      "Images"
    ],
    "Method": [
      "Movement",
      "Biometric",
      "AI"
    ],
    "IP": "192.0.2.1"
  },
  "Sensor": [
    {
      "IP": "192.0.2.2",
      "Name": "Camera #23",
      "Model": "SuperDuper Camera v1",
      "Location": "Hallway to server room A1"
    }
  ],
  "Source": [
    {
      "Note": "Black Organization, aka. APT 4869"
    }
  ],
  "Vector": [
    {
      "Category": ["Man"],
      "TI": ["Name:FBI-Wanted"],
      "Name": "John Doe",
      "Note": "Codename Vodka, known henchman for APT 4869",
      "Size": "Medium",
      "Location": "Hallway to server room A1",
      "Attachment": ["pic01", "wanted"]
    }
  ],
  "Attachment": [
    {
      "Name": "wanted",
      "FileName": "fbi-wanted-poster.jpg",
      "Size": 1234567,
      "Ref": ["https://www.fbi.gov/wanted/topten"],
      "ContentType": "image/jpg",
      "ContentEncoding": "base64",
      "Content": "..."
    },
    {
      "Name": "pic01",
      "Note": "Hi-res picture showing John Doe near server room A1",
      "ExternalURI": ["ftps://192.0.2.1/cam23/20210510165211.jpg"],
      "ContentType": "image/jpg"
    }
  ]
}
```

### A.2. Cyberattack

Listing 2 describes an incident related to a potential bruteforce attack against the "root" user account of the server at 192.0.2.2 and 2001:db8::/32.

```
{
  "Version": "2.0",
  "ID": "819df7bc-35ef-40d8-bbee-1901117370b2",
  "Description": "Potential bruteforce attack on root user account",
  "Severity": "Medium",
  "CreateTime": "2021-05-10T16:55:29.196408+00:00",
  "StartTime": "2021-05-10T16:55:29+00:00",
  "Category": [
    "Attempt.Login"
  ],
  "Analyzer": {
    "Name": "SIEM",
    "Hostname": "siem.example.com",
    "Type": "Cyber",
    "Model": "Prelude SIEM 5.2",
    "Category": [
      "SIEM",
      "LOG"
    ],
    "Data": [
      "Log"
    ],
    "Method": [
      "Monitor",
      "Signature"
    ],
    "IP": "192.0.2.1"
  },
  "Sensor": [
    {
      "IP": "192.0.2.5",
      "Name": "syslog",
      "Hostname": "www.example.com",
      "Model": "rsyslog 8.2110",
      "Location": "Server room A1, rack 10"
    }
  ],
  "Target": [
    {
      "IP": "192.0.2.2",
      "Hostname": "www.example.com",
      "Location": "Server room A1, rack 10",
      "User": "root"
    },
    {
      "IP": "2001:db8::/32",
      "Hostname": "www.example.com",
      "Location": "Server room A1, rack 10",
      "User": "root"
    }
  ]
}
```

### A.3.  Server outage

Listing 3 describes an incident where the webserver at "www.example.com" encountered some kind of failure condition resulting in an outage.

```
{
  "Version": "2.0",
  "ID": "819df7bc-35ef-40d8-bbee-1901117370b3",
  "Description": "A server did not reply to an ICMP ping request",
  "Severity": "Medium",
  "Status": "Incident",
  "Cause": "Unknown",
  "CreateTime": "2021-05-10T16:59:11.875209+00:00",
  "StartTime": "2021-05-10T16:59:11.875209+00:00",
  "Category": [
    "Availability.Outage"
  ],
  "Analyzer": {
    "Name": "NMS",
    "Hostname": "nms.example.com",
    "Type": "Availability",
    "Model": "Vigilo NMS 5.2",
    "Category": [
      "NMS"
    ],
    "Data": [
      "Network"
    ],
    "Method": [
      "Monitor"
    ],
    "IP": "192.0.2.1"
  },
  "Target": [
    {
      "IP": "192.168.1.2",
      "Hostname": "www.example.com",
      "Service": "website",
      "Location": "Server room A1, rack 10"
    }
  ]
}
```

### A.4.  Combined incident

Listing 4 describes a combined incident resulting from the correlation of the previous physical, cyber and availability incidents.

```
{
  "Version": "2.0",
  "ID": "819df7bc-35ef-40d8-bbee-1901117370b4",
  "Description": "Intrusion and Sabotage detected",
  "Severity": "High",
  "Status": "Incident",
  "Cause": "Malicious",
  "CreateTime": "2021-05-10T16:59:15.075994+00:00",
  "StartTime": "2021-05-10T16:52:11+00:00",
  "Category": [
    "Intrusion.Burglary",
    "Attempt.Login",
    "Intrusion.SysCompromise",
    "Availability.Outage",
    "Availability.Sabotage",
    "Availability.Failure"
  ],
  "CorrelID": [
    "819df7bc-35ef-40d8-bbee-1901117370b1",
    "819df7bc-35ef-40d8-bbee-1901117370b2",
    "819df7bc-35ef-40d8-bbee-1901117370b3"
  ],
  "Analyzer": {
    "Name": "Correlator",
    "Hostname": "correlator.example.com",
    "Type": "Combined",
    "Model": "Unity 360 Hybrid Correlator v5.2",
    "Category": [
    ],
    "Data": [
      "Alert"
    ],
    "Method": [
      "Correlation"
    ],
    "IP": "192.0.2.1"
  },
  "Source": [
    {
      "Note": "Black Organization, aka. APT 4869"
    }
  ],
  "Vector": [
    {
      "Category": ["Man"],
      "TI": ["Name:FBI-Wanted"],
      "Name": "John Doe",
      "Note": "Codename Vodka, known henchman for APT 4869",
      "Size": "Medium"
    }
  ],
  "Target": [
    {
      "Location": "Server room A1"
    },
    {
      "IP": "192.0.2.2",
      "Hostname": "www.example.com",
      "User": "root"
    },
    {
      "IP": "192.0.2.2",
      "Hostname": "www.example.com",
      "Service": "website"
    }
  ]
}
```

## Appendix B. JSON Validation Schema (Non-normative)

Listing 5 contains a JSON Schema that can be used to validate incoming IDMEF messages prior to processing. Please note that extraneous linebreaks have been included due to formatting constraints.

FIXME: le type vectorCategoryEnum ne correspond pas a l'enumeration definie dans le document (voir remarque dans la classe Vector)

```
{
 "$schema": "http://json-schema.org/draft-04/schema#",
 "type": "object",
 "title": "IDMEF v2.0",
 "description": "JSON schema for IDMEF version 2",

 "definitions": {
  "attachmentNameType": {
   "description": "A unique identifier among attachments",
   "type": "string",
   "pattern": "^[a-zA-Z0-9]+$"
  },
  "observableNameType": {
   "description": "A unique identifier among observables",
   "type": "string",
   "pattern": "^[a-zA-Z0-9]+$"
  },
  "portType": {
   "description": "A network port number",
   "type": "integer",
   "minimum": 0,
   "maximum": 65535,
   "exclusiveMinimum": true
  },
  "timestampType": {
   "description": "A JSON string containing a timestamp (RFC 3339)",
   "type": "string",
   "pattern": "^[0-9]{4}-(0[0-9]|1[012])-([0-2][0-9]|3[01])T([0-1]
[0-9]|2[0-3]):[0-5][0-9]:([0-5][0-9]|60)(\\.[0-9]+)?(Z|[-+]([0-1]
[0-9]|2[0-3]):[0-5][0-9])?$"
  },
  "geoLocationType": {
   "description": "Geolocation coordinates (ISO 6709)",
   "type": "string",
   "pattern": "^[-+]?([0-9]+(\\.[0-9]*)?)(, ?[-+]?([0-9]+(\\.
[0-9]*)?)){1,2}$"
  },
  "unLocodeType": {
   "description": "A valid UN/LOCODE location (e.g. \"FR PAR\")",
   "type": "string",
   "pattern": "^[A-Z]{2} ?[A-Z]{3}$"
  },
  "ipAddressType": {
   "description": "An Internet Protocol address (version 4 or 6)",
   "type": "string",
   "pattern": "^(((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\\.){3}(25
[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)|([0-9a-fA-F]{1,4}:){7,7}[0-9a-f
A-F]{1,4}|([0-9a-fA-F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-f
A-F]{1,4}|([0-9a-fA-F]{1,4}:){1,5}(:[0-9a-fA-F]{1,4}){1,2}|([0-9a-f
A-F]{1,4}:){1,4}(:[0-9a-fA-F]{1,4}){1,3}|([0-9a-fA-F]{1,4}:){1,3}
(:[0-9a-fA-F]{1,4}){1,4}|([0-9a-fA-F]{1,4}:){1,2}(:[0-9a-fA-F]{1,4})
{1,5}|[0-9a-fA-F]{1,4}:((:[0-9a-fA-F]{1,4}){1,6})|:((:[0-9a-fA-F]
{1,4}){1,7}|:)|fe80:(:[0-9a-fA-F]{0,4}){0,4}%[0-9a-zA-Z]{1,}|::
(ffff(:0{1,4}){0,1}:){0,1}((25[0-5]|(2[0-4]|1{0,1}[0-9]){0,1}[0-9])
\\.){3,3}(25[0-5]|(2[0-4]|1{0,1}[0-9]){0,1}[0-9])|([0-9a-fA-F]{1,4}
:){1,4}:((25[0-5]|(2[0-4]|1{0,1}[0-9]){0,1}[0-9])\\.){3,3}(25[0-5]|
(2[0-4]|1{0,1}[0-9]){0,1}[0-9]))$"
  },
  "mediaTypeType": {
   "description": "A valid media type (RFC 7231)",
   "type": "string",
   "pattern": "^[-!#$%&'*+.^_`|~0-9a-zA-Z]+/[-!#$%&'*+.^_`|~0-9a-z
A-Z]+([ \t]*;[ \t]*[-!#$%&'*+.^_`|~0-9a-zA-Z]+=([-!#$%&'*+.^_`|~0-9
a-zA-Z]+|\"([]-~\t !#-[\\x80-\\xFF]|\\\\([\t 0-9a-zA-Z\\x80-\\xFF]
))*\"))*$"
  },
  "uuidType": {
   "description": "A Universally Unique IDentifier (RFC 4122)",
   "type": "string",
   "pattern": "^[0-9A-Fa-f]{8}(-[0-9A-Fa-f]{4}){3}-[0-9A-Fa-f]{12}$"
  },
  "protocolNameType": {
   "description": "A JSON string containing a service/protocol name",
   "type": "string",
   "pattern": "^[a-zA-Z0-9](-?[a-zA-Z0-9])*$"
  },
  "hashType": {
   "description": "A checksum (e.g. \"crc32:cbf43926\")",
   "type": "string",
   "pattern": "^[a-zA-Z0-9-]+:([a-fA-F0-9]{2})+$"
  },

  "statusEnum": {
   "description": "Possible alert statuses",
   "enum": [
    "Event",
    "Incident"
   ]
  },
  "causeEnum": {
   "description": "Possible alert causes",
   "enum": [
     "Normal",
     "Error",
     "Malicious",
     "Malfunction",
     "Natural",
     "Unknown"
```

```
 ]
},
"severityEnum": {
 "description": "Possible alert severities",
 "enum": [
  "Unknown",
  "Info",
  "Low",
  "Medium",
  "High"
 ]
},
"analyzerCategoryEnum": {
 "description": "Possible analyzer categories",
 "enum": [
  "1DLiS",
  "2DLiS",
  "3DLiS",
  "1DLaS",
  "2DLaS",
  "3DLaS",
  "VAD",
  "HAR",
  "FRC",
  "VNIR",
  "SWIR",
  "LWIR",
  "MWIR",
  "ADS",
  "ODC",
  "WEA",
  "DDOS",
  "SPAM",
  "AV",
  "EDR",
  "FW",
  "NIDS",
  "HIDS",
  "WIDS",
  "PROX",
  "WAF",
  "HPT",
  "LOG",
  "IAM",
  "VPN",
  "ETL",
  "RASP",
  "BAST",
  "NAC",
  "SIEM",
  "NMS"
 ]
},
"analyzerTypeEnum": {
 "description": "Possible analyzer types",
 "enum": [
  "Cyber",
  "Physical",
  "Availability",
  "Combined"
 ]
},
"analyzerDataEnum": {
 "description": "Possible types of data/sensors",
 "enum": [
  "Light",
  "Noise",
  "Touch",
  "Images",
  "Vibration",
  "Lidar",
  "Thermic",
  "Seismic",
  "Temperature",
  "Rain",
  "Water",
  "Humidity",
  "Particles",
  "Contact",
  "MagneticField",
  "Acoustics",
  "Fog",
  "External",
  "Reporting",
  "Connection",
  "Datagram",
  "Content",
  "Data",
  "File",
  "Flow",
  "Log",
  "Protocol",
  "Host",
  "Network",
  "Alert",
  "Relay",
  "Auth",
  "SNMP"
 ]
},
"analyzerMethodEnum": {
 "description": "Possible detection methods",
 "enum": [
  "Biometric",
  "Signature",
  "Monitor",
  "Policy",
  "Statistical",
  "AI",
  "Heat",
  "Movement",
  "Blackhole",
  "Heuristic",
  "Integrity",
  "Honeypot",
  "Tarpit",
  "Recon",
  "Correlation",
  "Threshold"
 ]
},
"vectorCategoryEnum": {
 "description": "Possible categories for attack vectors",
 "enum": [
  "Unknown",
  "Face",
  "RunningMan",
  "Human",
  "Man",
  "Woman",
  "Chilren",
  "Animal",
  "Object",
  "Blast",
  "Fire",
  "Wind",
  "Snow",
  "Rain",
  "Chemical",
  "Smoke",
  "Vapors",
  "Drug",
  "Device",
  "Drone",
  "Car",
  "Truck",
  "Vehicle",
  "Bird",
  "Storm",
  "HighTemperature",
  "Artifact",
  "AutonomousSystem",
  "Directory",
  "DomainName",
  "EmailAddress",
  "EmailMessage",
  "File",
  "IPv4Address",
  "IPv6Address",
  "Mutex",
  "NetworkTraffic",
  "Process",
  "URL",
  "UserAccount",
  "WindowsRegistryKey",
  "X509Certificate"
 ]
},
"vectorSizeEnum": {
 "description": "Possible sizes for attack vectors",
 "enum": [
  "Small",
  "Medium",
  "Large",
  "Huge"
 ]
},

"categoryEnum": {
 "description": "Possible alert categories",
 "enum": [
  "Abusive.Spam",
  "Abusive.Harassment",
  "Abusive.Illicit",
  "Malicious.System",
  "Malicious.Botnet",
  "Malicious.Distribution",
  "Malicious.Configuration",
  "Recon.Scanning",
  "Recon.Sniffing",
  "Recon.SocialEngineering",
  "Attempt.Exploit",
  "Attempt.Login",
  "Attempt.NewSignature",
  "Intrusion.AdminCompromise",
  "Intrusion.UserCompromise",
   "Intrusion.AppCompromise",
   "Intrusion.SysCompromise",
   "Intrusion.Burglary",
   "Availability.DoS",
   "Availability.DDoS",
   "Availability.Misconf",
   "Availability.Theft",
   "Availability.Sabotage",
   "Availability.Outage",
   "Availability.Failure",
   "Information.UnauthorizedAccess",
   "Information.UnauthorizedModification",
   "Information.DataLoss",
   "Information.DataLeak",
   "Fraud.UnauthorizedUsage",
   "Fraud.Copyright",
   "Fraud.Masquerade",
   "Fraud.Phishing",
   "Vulnerable.Crypto",
   "Vulnerable.DDoS",
   "Vulnerable.Surface",
   "Vulnerable.Disclosure",
   "Vulnerable.System",
   "Geophysical.Earthquake",
   "Geophysical.MassMovement",
   "Geophysical.Volcanic",
   "Meteorological.Temperature",
   "Meteorological.Fog",
   "Meteorological.Storm",
   "Hydrological.Flood",
   "Hydrological.Landslide",
   "Hydrological.Wave",
   "Climatological.Drought",
   "Climatological.LakeOutburst",
   "Climatological.Wildfire",
   "Biological.Epidemic",
   "Biological.Insect",
   "Biological.Animal",
   "Extraterrestrial.Impact",
   "Extraterrestrial.SpaceWeather",
   "Other.Uncategorized",
   "Other.Undetermined",
   "Test.Test"
  ]
 }
},

"required": [
 "Version",
 "ID",
 "CreateTime",
 "Analyzer"
],
"additionalProperties": false,
"properties": {
 "Version": {
  "description": "Version of the IDMEFv2 Format",
  "enum": ["2.0.3"]
 },
 "ID": {
  "description": "128-bit Universally Unique IDentifier (UUID)",
  "$ref": "#/definitions/uuidType"
 },
 "Entity": {
  "description": "Tenant identifier to support multi-tenancy",
  "type": "string"
 },
 "Category": {
  "description": "The ENISA:RIST incident category & subcategory",
  "type": "array",
  "items": {
   "$ref": "#/definitions/categoryEnum"
  }
 },
 "Cause": {
  "description": "Alert cause's origin",
  "$ref": "#/definitions/causeEnum"
 },
 "Description": {
  "description": "Short free text human-readable description",
  "type": "string"
 },
 "Status": {
  "description": "Alert state in the overall alert lifecycle",
  "$ref": "#/definitions/statusEnum"
 },
 "Severity": {
  "description": "Severity of the alert",
  "$ref": "#/definitions/severityEnum"
 },
 "Confidence": {
  "description": "Confidence in detection",
  "type": "number",
  "minimum": 0,
  "maximum": 1
 },
 "Note": {
  "description": "Free text human-readable additional note",
  "type": "string"
 },
 "CreateTime": {
  "description": "Message creation timestamp",
  "$ref": "#/definitions/timestampType"
 },
 "StartTime": {
  "description": "Deduced start of the event",
  "$ref": "#/definitions/timestampType"
 },
 "CeaseTime": {
  "description": "Deduced end of the event",
  "$ref": "#/definitions/timestampType"
 },
 "DeleteTime": {
  "description": "Message deletion timestamp",
  "$ref": "#/definitions/timestampType"
 },
 "AltNames": {
  "description": "Alternative identifiers",
  "type": "array",
  "items": {
   "type": "string"
  }
 },
 "AltCategory": {
  "description": "Alternative categories",
  "type": "array",
  "items": {
   "type": "string"
  }
 },
 "Ref": {
  "description": "References related to the alert",
  "type": "array",
  "items": {
   "type": "string",
   "format": "uri"
  }
 },
 "CorrelID": {
  "description": "Messages used to create this message",
  "type": "array",
  "items": {
   "$ref": "#/definitions/uuidType"
  }
 },
 "AggrCondition": {
  "description": "Conditions used to aggregate messages",
  "type": "array",
  "items": {
   "type": "string"
  }
 },
 "PredID": {
  "description": "Previous messages which are now obsolete",
  "type": "array",
  "items": {
   "$ref": "#/definitions/uuidType"
  }
 },
 "RelID": {
  "description": "Other messages related to this message",
  "type": "array",
  "items": {
   "$ref": "#/definitions/uuidType"
  }
 },

 "Analyzer": {
  "description": "Analyzer from which the message originates",
  "type": "object",
  "required": [
   "IP",
   "Name",
   "Model",
   "Category",
   "Data",
   "Method"
  ],

  "additionalProperties": false,
  "properties": {
   "IP": {
    "description": "IP address",
    "$ref": "#/definitions/ipAddressType"
   },
   "Name": {
    "description": "Name of the analyzer",
    "type": "string"
   },
   "Hostname": {
    "description": "Hostname of this analyzer",
    "type": "string"
   },
   "Type": {
    "description": "Analyzer type",
    "$ref": "#/definitions/analyzerTypeEnum"
   },
   "Model": {
    "description": "Generic name, brand, version",
    "type": "string"
   },
   "Category": {
    "description": "Analyzer categories",
    "type": "array",
    "items": {
     "$ref": "#/definitions/analyzerCategoryEnum"
    }
   },
   "Data": {
    "description": "Data used during the detection",
    "type": "array",
    "items": {
     "$ref": "#/definitions/analyzerDataEnum"
    }
   },
   "Method": {
    "description": "Detection method",
    "type": "array",
    "items": {
     "$ref": "#/definitions/analyzerMethodEnum"
    }
   },
   "GeoLocation": {
    "description": "GPS coordinates for the analyzer",
    "$ref": "#/definitions/geoLocationType"
   },
   "UnLocation": {
    "description": "Standard UN/LOCODE location",
    "$ref": "#/definitions/unLocodeType"
   },
   "Location": {
    "description": "Internal location of the analyzer",
    "type": "string"
   }
  }
 },

 "Sensor": {
  "type": "array",
  "items": {
   "description": "Sensor(s) used by the analyzer for its analysis",
   "type": "object",
   "required": [
    "IP",
    "Name",
    "Model"
   ],

   "additionalProperties": false,
   "properties": {
    "IP": {
     "description": "The sensor's IP address",
     "$ref": "#/definitions/ipAddressType"
    },
    "Name": {
     "description": "Name of the sensor",
     "type": "string"
    },
    "Hostname": {
     "description": "Hostname of the sensor",
     "type": "string"
    },
    "Model": {
     "description": "Generic name, brand, version",
     "type": "string"
    },
    "UnLocation": {
     "description": "Standard UN/LOCODE location",
     "$ref": "#/definitions/unLocodeType"
    },
    "Location": {
     "description": "Internal location of the sensor",
     "type": "string"
    },
    "CaptureZone": {
     "description": "Sensor capture zone (as serialized JSON)",
     "type": "string"
    }
   }
  }
 },

 "Source": {
  "type": "array",
  "items": {
   "description": "Possible source(s) of the event",
   "type": "object",

   "additionalProperties": false,
   "properties": {
    "UnLocation": {
     "description": "Standard UN/LOCODE location for this source",
     "$ref": "#/definitions/unLocodeType"
    },
    "Location": {
     "description": "Internal location (for internal sources)",
     "type": "string"
    },
    "GeoLocation": {
     "description": "GPS coordinates for the source",
     "$ref": "#/definitions/geoLocationType"
    },
    "Note": {
     "description": "Free text human-readable additional note",
     "type": "string"
    },
    "TI": {
     "description": "Threat Intelligence about the source",
     "type": "array",
     "items": {
      "type": "string"
     }
    },
    "IP": {
     "description": "Source IP address",
     "$ref": "#/definitions/ipAddressType"
    },
    "Hostname": {
     "description": "Hostname of this source",
     "type": "string"
    },
    "User": {
     "description": "User ID or login responsible for the alert",
     "type": "string"
    },
    "Email": {
     "description": "Email address",
     "type": "string",
     "format": "email"
    },
    "Protocol": {
     "description": "Protocols in connections from/to this source",
     "type": "array",
     "items": {
      "$ref": "#/definitions/protocolNameType"
     }
    },
    "Port": {
     "description": "Source ports involved",
     "type": "array",
     "items": {
      "$ref": "#/definitions/portType"
     }
```

```
   },
   "Attachment": {
    "description": "Attachments related to this source",
    "type": "array",
    "items": {
     "$ref": "#/definitions/attachmentNameType"
    }
   },
   "Observable": {
    "description": "Observables related to this source",
    "type": "array",
    "items": {
     "$ref": "#/definitions/observableNameType"
    }
   }
  }
 }
},

"Target": {
 "type": "array",
 "items": {
  "description": "Possible target(s) of the event",
  "type": "object",

  "additionalProperties": false,
  "properties": {
   "UnLocation": {
    "description": "Standard UN/LOCODE location for this target",
    "$ref": "#/definitions/unLocodeType"
   },
   "Location": {
    "description": "Internal location of the target",
    "type": "string"
   },
   "GeoLocation": {
    "description": "GPS coordinates for the target",
    "$ref": "#/definitions/geoLocationType"
   },
   "Note": {
    "description": "Free text human-readable additional note",
    "type": "string"
   },
   "IP": {
    "description": "Target IP address",
    "$ref": "#/definitions/ipAddressType"
   },
   "Hostname": {
    "description": "Hostname of this target",
    "type": "string"
   },
   "Service": {
    "description": "Impacted service/process",
    "type": "string"
   },
   "User": {
    "description": "User ID or login targeted by the alert",
    "type": "string"
   },
   "Email": {
    "description": "Email address",
    "type": "string"
   },
   "Port": {
    "description": "Ports affected on this target",
    "type": "array",
    "items": {
     "$ref": "#/definitions/portType"
    }
   },
   "Attachment": {
    "description": "Attachments related to this target",
    "type": "array",
    "items": {
     "$ref": "#/definitions/attachmentNameType"
    }
   },
   "Observable": {
    "description": "Observables related to this target",
    "type": "array",
    "items": {
     "$ref": "#/definitions/observableNameType"
    }
   }
  }
 }
},

"Vector": {
 "type": "array",
 "items": {
  "description": "Vector(s) of the event",
  "type": "object",
  "required": [
   "Category"
  ],

  "additionalProperties": false,
  "properties": {
   "Category": {
    "description": "Category for the detected \"vector\"",
    "type": "array",
    "items": {
     "$ref": "#/definitions/vectorCategoryEnum"
    }
   },
   "TI": {
    "description": "Threat Intelligence about the vector",
    "type": "array",
    "items": {
     "type": "string"
    }
   },
   "Name": {
    "description": "Name of the detected vector or \"Unknown\"",
    "type": "string"
   },
   "Size": {
    "description": "Average size of the detected vector",
    "$ref": "#/definitions/vectorSizeEnum"
   },
   "UnLocation": {
    "description": "UN Location of the vector",
    "$ref": "#/definitions/unLocodeType"
   },
   "GeoLocation": {
    "description": "GPS coordinates for the vector",
    "$ref": "#/definitions/geoLocationType"
   },Acknowledgments
   "GeoRadius": {
    "description": "Error margin in meters",
    "type": "number"
   },
   "Location": {
    "description": "Internal location",
    "type": "string"
   },
   "Note": {
    "description": "Free text human-readable additional note",
    "type": "string"
   },
   "Attachment": {
    "description": "Attachments related to this vector",
    "type": "array",
    "items": {
     "$ref": "#/definitions/attachmentNameType"
    }
```

```
   },
   "Observable": {
    "description": "Observables related to this vector",
    "type": "array",
    "items": {
     "$ref": "#/definitions/observableNameType"
    }
   }
  }
 }
},

"Attachment": {
 "type": "array",
 "items": {
  "description": "Data linked to a source, target or vector",
  "type": "object",
  "required": [
   "Name"
  ],

  "additionalProperties": false,
  "properties": {
   "Name": {
    "description": "Unique identifier among attachments",
    "$ref": "#/definitions/attachmentNameType"
   },
   "FileName": {
    "description": "Attachment filename",
    "type": "string"
   },
   "Hash": {
    "description": "Checksum of the attachment's content",
    "type": "array",
    "items": {
     "$ref": "#/definitions/hashType"
    }
   },
   "Size": {
    "description": "Content length (in bytes)",
    "type": "integer"
   },
   "Ref": {
    "description": "Link to information about this attachment",
    "type": "array",
    "items": {
     "type": "string",
     "format": "uri"
    }
   },
   "ExternalURI": {
    "description": "Link to external copies (e.g. online copies)",
    "type": "array",
    "items": {
     "type": "string",
     "format": "uri"
    }
   },
   "Note": {
    "description": "Free text human-readable additional note",
    "type": "string"
   },
   "ContentType": {
    "description": "Media Type of the attachment (RFC 2046)",
    "$ref": "#/definitions/mediaTypeType"
   },
   "ContentEncoding": {
    "description": "Content encoding",
    "type": "string"
   },
   "Content": {
    "description": "The attachment's content (if embedded)",
    "type": "string"
   }
  }
 }
},

"Observable": {
 "type": "array",
 "items": {
  "description": "Metadata linked to a source, target or vector",
  "type": "object",
  "required": [
   "Name",
   "Content"
  ],

  "additionalProperties": false,
  "properties": {
   "Name": {
    "description": "Unique identifier among observables",
    "$ref": "#/definitions/observableNameType"
   },
   "Reference": {
    "description": "Reference to the observable's specification",
    "type": "string"
   },
     "Content": {
      "description": "Observable content",
      "type": "string"
     }
    }
   }
  }
 }
}
```

## Authors' Addresses

Gilles Lehmann
Telecom Sud Paris
France

Email: gilles.lehmann@telecom-sudparis.eu

Thomas Andrejak
CS GROUP
France

Email: thomas.andrejak@csgroup.eu

Francois Poirotte
CS GROUP
France

Email: francois.poirotte@csgroup.eu