Tomáš Čejka
cejkat@cesnet.cz

Differences between IODEF and IDEA

JSON representation

IETF94 MILE WG meeting

Outline

Do we want IODEF in JSON?

Aim of this presentation

- Brief description of IODEF purpose & characterization
- Brief description of IDEA purpose & characterization (designed in CESNET for incident information sharing)
- Comparison of examples from RFC5070-bis IODEF2 and IDEA
- Summary of differences

IODEF Motivation/Characterization I

Taken from RFC5070-bis:

Section 1.4

> *"The data model serves as a transport format. Therefore, its specific representation is not the optimal representation for* ***on-disk storage****,* ***long-term archiving****, or* ***in-memory processing****."*

Section 1.5

The section defines XML as the only representation.

IODEF Motivation/Characterization II

IODEF in general / as I understand IODEF

- IODEF is a human-readable and human-processable representation of incident information.
- IODEF tries hard to describe everything from the real world.
- Information about incident can be described in multiple ways.
- Information can be placed on more than one place.

(details on the following slides)

IDEA Motivation/Characterization I

IDEA in general

- Primary for machine processing of event description.
- Shallow structure without recursion.
- "incident-based" describes only incident's technical environment
  (not incident handling or social environment)
- *Source* (of incident) is always evil, *Target* is a victim.
- IDEA represents just incident reports, it does not take into consideration human processing or institutional processes.

Examples of IDEA:
https://csirt.cesnet.cz/en/idea/examples

```
{
   "Format": "IDEA0",
   "ID": "3ad275e3-559a-45c0-8299-6807148ce157",
   "DetectTime": "2014-03-22T10:12:56Z",
   "Category": ["Recon.Scanning"],
   "ConnCount": 633,
   "Description": "Ping scan",
   "Source": [{
         "IP4": ["93.184.216.119"],
         "Proto": ["icmp"]
      }],
   "Target": [{
         "Proto": ["icmp"],
         "IP4": ["147.32.3.0/24"],
         "Anonymised": true
      }]
}
```

Practical Differences Using Examples

The whole examples can be found in
draft-cejkat-mile-iodef-and-idea-00

https://datatracker.ietf.org/doc/draft-cejkat-mile-iodef-and-idea/

Comparison of IODEF and IDEA — Examples I

Aim of document

IODEF:

```
<Incident purpose="reporting">
```

IDEA:
Every IDEA message is an event report.

Classification of events

IODEF:

```
<Impact completion="failed" type="admin"/>
```

IDEA:
When completion “fails”, it means an “attempt” in IDEA.

Comparison of IODEF and IDEA — Examples II

Representation of Contact information

IODEF:

```
<Contact role="creator" type="organization">
```

IDEA:

Expression of Contact is very limited in IDEA. The reason is that information about human (non-technical) environment as well as organizational relations are not used for machine processing. However, there is a way how to represent one instance of `<ContactName>`, `<RegistryHandle>` or `<Email>` in IDEA.

Comparison of IODEF and IDEA — Examples III

```
"Node": [{
  "Name": "com.example.csirt.scandetector",
  "Ref": [
    "urn:mailto:contact@csirt.example.com",
    "urn:tel:+1 412 555 12345"
  ],
  "Note": "Example.com CSIRT scan detector"
}]
```

Comparison of IODEF and IDEA — Examples IV

Who is Source?

IODEF: It seems to be network flow oriented:

```
<System category="source"><Node>...</Node></System>
<System category="target"><Node>...</Node></System>
```

IDEA:

Source is always “evil” — it is e.g. an infected entity, a source of infection, an attacker. Source need not to be a technical source (such as origin of network flow, source address of packet). Source is suitable for mitigation or blacklisting.

Comparison of IODEF and IDEA — Examples V

Representation of history

IODEF:

```
<History>...</History>
```

IDEA:

History is not described at all.

Confidentiality

IODEF:

```
<Contact role="tech" type="person"
    restriction="need-to-know">
```

IDEA:

Confidentiality/Restriction is handled by Traffic Light Protocol (TLP) for the whole IDEA message. IDEA messages contain only information that a receiver can read and use.

Comparison of IODEF and IDEA — Examples VI

Receiver's actions

IODEF:

```
<Incident purpose="mitigation">
<Expectation action="contact-sender">
<Expectation action="investigate">
<Expectation action="block-host">
```

IDEA:

IDEA messages do not specify expected action or reply. Parties that use IDEA can agree on format of indication of possible action. However, actions are up to receiver. Expectation *block-host/investigate* is not covered — (human tasks)

Comparison of IODEF and IDEA — Examples VII

How to describe severity?

IODEF:

```
<Impact type="dos" severity="high" />
```

IDEA:

IDEA has no metrics to specify severity. It is difficult to specify a common scale for different entities and different incident types.

Representation of rate counters

IODEF:

```
<Counter type="byte" duration="second">10000</Counter>
```

IDEA:

Incident in IDEA must be represented in exact time frames (WinStartTime, WinEndTime). Counters are related to the time frame.

Comparison of IODEF and IDEA — Examples VIII

```
{
  "Format": "IDEA0",
  "WinStartTime": "2006-06-08T01:01:02-05:00",
  "WinEndTime": "2006-06-08T01:06:02-05:00",
  "ByteCount": 260000,
  ...
  "Source": [{
    "ByteCount": 10000,
    ...
    }, {...}],
}
```

Comparison of IODEF and IDEA — Examples IX

What related information to include?

IODEF:

```
<System category="intermediate">
```

IDEA:

IDEA describes only one fact/event/incident per message.

Example in draft-cejkat-mile-iodef-and-idea-01, section 2.6

Summary of Differences

IDEA

- Shallow structure, information should be on one place.
- Does not cover everything from IODEF.
- Because it is designed for different purposes — storage, machine (automatic) processing.
- IDEA represents information from IDSs etc — several messages per hour, it must be processed automatically.

IODEF

- Data representation for humans, who can "understand".
- General enough to represent almost everything, it can contain free-form text information.
- This brings a complexity of an IODEF document structure → difficult for machine processing.

Conclusion

Results of our analysis

- JSON version of the format should be built from the grounds up and take into consideration JSON specifics. Straightforward XML to JSON translation would lead to cumbersome result.
- IDEA is a not suitable equivalent:
    - it represents a subset of IODEF (EventData),
    - it is designed with different purpose (storage, machine oriented).
- In practice, both formats are needed: human processing, machine processing.
- IDEA can be used/embedded for JSON-based IODEF.

Questions for discussion about JSON Format

- Who/what will work with JSON format?
- How to create a JSON representation?
- Can be IDEA used as an inspiration?
- What should be the next steps?
- How to continue?
- Should it be a separated working group?