JSON binding of IODEF

draft-takahashi-mile-jsoniodef-00.txt

Takeshi Takahashi, NICT

Agenda

1. Why we want JSON bindings of IODEF?
2. Our approach and current status
3. Discussion issues

Depending on use cases, JSON is preferred to XML

Characteristics of XML and JSON (my personal opinion)

XML: structured texts

- Is expressive and flexible
- But is heavy for parsing, redundant, wordy, complex, and large by size

JSON: structured data

- Is simple and easy to define the data types, and light for parsing
- But is not necessarily designed to be long-term archive

Depending on the use cases, the preferred method may differ

- XML document may be preferred for adding metadata to existing text-based document
- JSON document may be preferred by program. Program may import/export and manipulate JSON document

Here is our use case. We have an alerting system

NICTER system overview

System Overview	• NICTER is a system for darknet traffic monitoring and produces security alerts automatically • See http://www.nicter.jp for more information
User	• More than 500 organizations in JP • Organizations in more than 10 countries
Issue	• Use standardized formats for alerts • Make the alerts usable for the programs receiving them (for automated security operations incl. triage)

FYI: an example alert of our system (1/2)

We have several representations, but the one in XML is as follows

```
<?xml version="1.0"?>
<NicterEvent>
  <Header><EventType>DaedalusAlert</EventType>
            <CreateTime>2016-06-01 18:15:18</CreateTime></Header>
  <DaedalusAlertHeader>
    <AlertID>13353</AlertID> <OrgID>7</OrgID> <Trigger>Periodic</Trigger> <Duration>900</Duration>
  </DaedalusAlertHeader>
  <AlertData EventTime="2016-06-01 18:05:33" EventID="186995" SrcIP="192.228.139.118" SrcCC="JP"
    TotalPacketCount="3" DisplayedPacketCount="3" Type="Continued">
      <Packet PacketTime="2016-06-01 18:05:24" DstIP="" DstCC="" DstPort="23" SrcPort="49183"
        Protocol="6" Flag="2" DarknetType="external"/>
      <Packet PacketTime="2016-06-01 18:05:27" DstIP="" DstCC="" DstPort="23" SrcPort="49183"
        Protocol="6" Flag="2" DarknetType="external"/>
      <Packet PacketTime="2016-06-01 18:05:33" DstIP="" DstCC="" DstPort="23" SrcPort="49183"
        Protocol="6" Flag="2" DarknetType="external"/>
    </AlertData>
</NicterEvent>
```

FYI: an example alert of our system (2/2)

We currently prefer simple text description to XML

It is much more simple, and easy to read.

Nevertheless, programs may find it troublesome to use this data

```
AlertType: Periodic
==============================
Date            : 2016-06-01 18:05:33
EventID         : 186995
SrcIPAddress    : 192.228.139.118
TotalPackets    : 3
------------------------------
Date          : 2016-06-01 18:05:24
Protocol      : TCP
Flow          : 192.228.139.118:49183 -> (masked):23
Flag          : 2
DarknetType   : external
---------
Date          : 2016-06-01 18:05:27
Protocol      : TCP
Flow          : 192.228.139.118:49183 -> (masked):23
Flag          : 2
DarknetType   : external
---------
Date          : 2016-06-01 18:05:33
Protocol      : TCP
Flow          : 192.228.139.118:49183 -> (masked):23
Flag          : 2
DarknetType   : external
---------
```

Our use case prefers JSON binding

The rich capability of XML is just not necessary for this system

1. The alerts are kept simple and short, and won't be complex
2. Flexibility is not important.

Our data is suitable for JSON to be represented

1. Simple data may prefer JSON binding
2. JSON is good at representing data structure concisely

Receiver of the alerts can easily process the data by program

1. JSON object is easy to handle by program (data structure can be understood without reading the schema)
2. Programs at receiver side can use the object for automating security operations

Agenda

1. Why we want JSON binding of IODEF?
2. Our approach and current status
3. Discussion issues

JSON representation should be simple for readers and easy for IODEF document creators

Maintain compatibility

1. IODEFv2 in XML should be convertible into JSON
2. Its expressiveness should not increase
3. Consider compatibility with STIX-related specs
4. We'll prepare some tools to cope with the above issues

Facilitate its description

1. Name of the elements could be changed a bit
   to facilitate the creator of the JSON document
   (e.g., Port -> Portlist to represent that the variable is an array)
2. Some simplified expression could be supported
   (e.g. the description of IP address and port)

FYI: an example alert using JSON that is directly converted from IODEFv2 in XML


```
{
  "version": "2.0", "lang": "en",  "Incidents": [
    {
      "IncidentID": {
        "id": "13353",
        "name": "alert.daedalus.nict.go.jp"
      },
      "EventData": [
        {
          "ReportTime": "2016-06-01 18:05:33",
          "System": {
            "category": "source",
            "Node": {
              "Address": {
                "category": "ipv4-addr",
                "AddressValue": "192.228.139.118"
              },
              "Location": "OrgID=7"
            },
            "Service": {
              "ip-protocol": "6",
              "Port": "49183"
            }
          },
"EventData": {
        "ReportTime": "2016-06-01 18:05:24",
        "System": {
          "category": "target",
          "Node": {},
          "Service": {
            "Port": "23"
          }
        }
      },
      "EventData": {
        "ReportTime": "2016-06-01 18:05:27",
        "System": {
          "category": "target",
          "Node": {},
          "Service": {
            "Port": "23"
          }
        }
      },
"EventData": {
        "ReportTime": "2016-06-01 18:05:33",
        "System": {
          "category": "target",
          "Node": {},
          "Service": {
            "Port": "23"
          }
        }
      }
    ],
    "GenerationTime": "2016-06-01 18:15:18",
    "Contacts": [],
    "purpose": "reporting"
  }
 ]
}
```


It is still very complicated.
Direct conversion is not enough.

Agenda

1. Why we want JSON binding of IODEF?
2. Our approach and current status
3. Discussion issues

Questions

1. What will be the best way for defining the JSON representation? JSON schema? Any other options?

2. Anybody interested in being a co-author?

3. Do we want to work on this within MILE?