JSON binding of IODEF

draft-takahashi-mile-jsoniodef-00.txt

Takeshi Takahashi, NICT

Agenda

1. Summary of discussion until now
2. Difference from the original IODEF version 2
3. Question
4. Next steps

Summary of discussion until now

1. People are generally in favor of defining the JSON binding

2. The difference betwee JSON IODEF and IODEFv2 was unclear
   - "*Is it going to be a subset? or extension?*"
   - "*Keep in mind that XML and JSON contains the same level of expressiveness*"

3. [Local discussion] JSON-friendly adjustment should be made
   - "*JSON is often used by programs. It does not need all fields of IODEF. When we want to use all fields, we prefer to use XML (=IODEF v2 XML).*"
   - "*The concept of IODEFv2 is maximum flexibility. The JSON version may have different concept.*"
   - "*IODEFv2 can carry various data, but it is not so easy (it is verbous) to write these data in IODEFv2 (XML)* "

How different is the draft compared to the IODEF v2?

Compatible

1. The JSON IODEF is perfectly compatible with IODEFv2
2. There is no mandatory field for JSON IODEF
3. The JSON IODEF cannot express the type of data that could not be expressed in IODEFv2

Extended (for better usability for JSON)

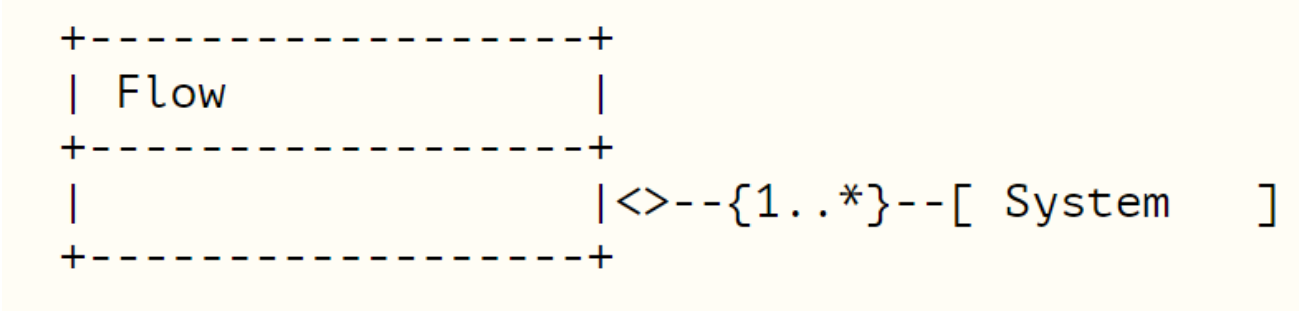
1. Some element names were changed
2. Some classes that exist only for semantical consistency is omitted
3. Some simplified expression is permitted
4. Profile is prepared

Some element names were changed

1. Use “-list” to the field that is an array (e.g., Port -> Portlist, EventData -> EventDataList). In this way, we can know that the field is an array without looking up the schema.

Some classes that exist only for semantical consistency is omitted

1. Flow class (a container for System classes)
2. ApplicationHeader class (a container for ApplicationHeaderField classes)
3. SignatureData class (a container for ds:Signature classes)
4. IndicatorData class (a container for Indicator classes)



Some simplified expression is permitted

1. Some simplified expression is permitted (e.g. “133.243.22.34:80”)

Profile is prepared

1. Profile provides a means to limit the use of IODEF classes for different use cases
   - Profile allows us to select the field we use from IODEFv2
   - JSON IODEF does not have any mandatory field, but profile may be used to put some restrictions
   - It is often the case that the programs wish to use only limited field of IODEFv2. Thus using profile would be helpful

2. Only one profile is described in this document (our own use case)

3. Arbitrary, user-defined profile is supported

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"
            },
          },
          "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.

FYI: an example alert using JSON IODEF


```
{
  "version": "2.0",
  "Incidents": [ {
     "IncidentID": {
       "id": "13353",
       "name": "alert.daedalus.nict.go.jp"
     },
     "EventDataList": [ {
        "ReportTime": "2016-06-01 18:05:33",
        "System": {
          "category": "source",
          "Node": "192.228.139.118:49183",
        },
        "EventData": {
          "ReportTime": "2016-06-01 18:05:24",
          "System": {
            "category": "target",
            "Node": ":23"
          }
        },
"EventData": {
          "ReportTime": "2016-06-01 18:05:27",
          "System": {
            "category": "target",
            "Node": ":23"
          }
        },
        "EventData": {
          "ReportTime": "2016-06-01 18:05:33",
          "System": {
            "category": "target",
            "Node": ":23"
          }
        }
      }
     ],
     "GenerationTime": "2016-06-01 18:15:18",
     "purpose": "reporting"
   }
  ]
}
```

(tags and line breaks are modified)


```
{
  "version": "2.0",
  "Incidents": [ {
      "IncidentID": { "id": "13353", "name": "alert.daedalus.nict.go.jp"},
      "EventDataList": [ {
          "ReportTime": "2016-06-01 18:05:33",
          "System": { "category": "source", "Node": "192.228.139.118:49183",},
          "EventData": { "ReportTime": "2016-06-01 18:05:24", "System": { "category": "target", "Node": ":23"}},
          "EventData": { "ReportTime": "2016-06-01 18:05:27", "System": { "category": "target", "Node": ":23"}},
          "EventData": { "ReportTime": "2016-06-01 18:05:33", "System": { "category": "target", "Node": ":23"}}
        }
      ],
      "GenerationTime": "2016-06-01 18:15:18",
      "purpose": "reporting"
    }
  ]
}
```


This form is easier for our system to process the data, even easier for us to read

A question to the WG

I want to upload new version to the DataTracker, but…

Can I upload the new version as the WG draft?

- We have done a poll over the mailing list
- Though no decision was declared, we have seen several supports, no objection was raised

Next steps

Publish next version of the draft that contains all the content (no placeholder)