JSON binding of IODEF

draft-ietf-mile-jsoniodef-03.txt
https://github.com/milewg/draft-ietf-mile-jsoniodef

Takeshi Takahashi, Roman Danyliw, and Mio Suzuki
Current status of the draft

1. Based on the discussion at the IETF 100 in Singapore, the draft was revised

2. We made the draft more concise. (60 pages -> 39 pages)
This draft is now much more concise than before

Table of contents (of the draft)

1. Introduction
2. IODEF Data Types
3. IODEF JSON Data Model
4. Examples
5. The IODEF Data Model (JSON Schema)
6. Acknowledgements
7. IANA Considerations
8. Security Considerations
9. Normative References
Authors' Addresses

Volume reduced:
60 pages -> 39 pages
## IODEF Data Types

<table>
<thead>
<tr>
<th>IODEF Data Type</th>
<th>[RFC7970] Reference</th>
<th>JSON Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGER</td>
<td>Section 2.1</td>
<td>&quot;integer&quot; per [jsonschema]</td>
</tr>
<tr>
<td>REAL</td>
<td>Section 2.2</td>
<td>&quot;number&quot; per [jsonschema]</td>
</tr>
<tr>
<td>CHARACTER</td>
<td>Section 2.3</td>
<td>&quot;string&quot; per [jsonschema]</td>
</tr>
<tr>
<td>STRING</td>
<td>Section 2.3</td>
<td>&quot;string&quot; per [jsonschema]</td>
</tr>
<tr>
<td>ML_STRING</td>
<td>Section 2.4</td>
<td>see Section 2.2.1</td>
</tr>
<tr>
<td>BYTE</td>
<td>Section 2.5.1</td>
<td>&quot;string&quot; per [jsonschema]</td>
</tr>
<tr>
<td>BYTE[]</td>
<td>Section 2.5.1</td>
<td>&quot;string&quot; per [jsonschema]</td>
</tr>
<tr>
<td>HEXBIN</td>
<td>Section 2.5.2</td>
<td>&quot;string&quot; per [jsonschema]</td>
</tr>
<tr>
<td>HEXBIN[]</td>
<td>Section 2.5.2</td>
<td>&quot;string&quot; per [jsonschema]</td>
</tr>
<tr>
<td>ENUM</td>
<td>Section 2.6</td>
<td>&quot;enum&quot; array per [jsonschema]</td>
</tr>
<tr>
<td>DATETIME</td>
<td>Section 2.7</td>
<td>&quot;string&quot; per [jsonschema]</td>
</tr>
<tr>
<td>TIMEZONE</td>
<td>Section 2.8</td>
<td>&quot;string&quot; per [jsonschema]</td>
</tr>
<tr>
<td>PORTLIST</td>
<td>Section 2.9</td>
<td>&quot;string&quot; per [jsonschema]</td>
</tr>
<tr>
<td>POSTAL</td>
<td>Section 2.10</td>
<td>&quot;string&quot; per [jsonschema]</td>
</tr>
<tr>
<td>POSTAL_ML</td>
<td>Section 2.10</td>
<td>see ML_STRING, Section 2.2.1</td>
</tr>
<tr>
<td>PHONE</td>
<td>Section 2.11</td>
<td>&quot;string&quot; per [jsonschema]</td>
</tr>
<tr>
<td>EMAIL</td>
<td>Section 2.12</td>
<td>&quot;string&quot; per [jsonschema]</td>
</tr>
<tr>
<td>URL</td>
<td>Section 2.13</td>
<td>&quot;string&quot; per [jsonschema]</td>
</tr>
<tr>
<td>ID</td>
<td>Section 2.14</td>
<td>&quot;string&quot; per [jsonschema]</td>
</tr>
<tr>
<td>IDREF</td>
<td>Section 2.14</td>
<td>&quot;string&quot; per [jsonschema]</td>
</tr>
<tr>
<td>SOFTWARE</td>
<td>Section 2.15</td>
<td>see Section 2.2.2</td>
</tr>
<tr>
<td>STRUCTURED</td>
<td>RFC 7213</td>
<td>see Section 2.2.3</td>
</tr>
<tr>
<td>EXTENSION</td>
<td>Section 2.16</td>
<td>see Section 2.2.4</td>
</tr>
</tbody>
</table>

![Figure 1](image-url)
## IODEF JSON Data Model

<table>
<thead>
<tr>
<th>IODEF Class</th>
<th>Class Elements and Attribute</th>
<th>Corresponding Section in [RFC7970]</th>
</tr>
</thead>
</table>
Mapping between JSON and XML IODEF (Sec 3.2)

1. This document treats attributes and elements of each class defined in [RFC7970] equally and is agnostic on the order of their appearances.
2. Flow class is deleted, and classes with its instances now directly have instances of EventData class that used to belong to the Flow class.
3. ApplicationHeader class is deleted, and classes with its instances now directly have instances of ApplicationHeaderField class that used to belong to the ApplicationHeader class.
4. SignatureData class is deleted, and classes with its instances now directly have instance of Signature class that used to belong to the SignatureData class.
5. IndicatorData class is deleted, and classes with its instances now directly have the instances of Indicator class that used to belong to the IndicatorData class.
6. ObservableReference class is deleted, and classes with its instances now directly have uid-ref as an element.
7. Record class is deleted, and classes with its instances now directly have the instances of RecordData class that used to belong to the Record class.
8. The elements of ML_STRING type are prepared as two separate elements: one of STRING type and another of ML_STRING type, in order to maintain the simplicity of IODEF documents when writing with only STRING type characters.
JSON Schema is provided, but an issue exists

5. The IODEF Data Model (JSON Schema)

```json
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "definitions": {
      "action": {
         "enum": [
            "nothing",
            "contact-source-site",
            "contact-target-site",
            "contact-sender",
            "investigate",
            "block-host",
            "block-network",
            "block-port",
            "rate-limit-host",
            "rate-limit-network",
            "rate-limit-port",
            "redirect-traffic",
            "honeypot",
            "upgrade-software",
            "rebuild-asset",
            "harden-asset",
            "remediate-other",
            "status-triage",
            "status-new-info",
            "watch-and-report",
            "training",
            "defined-coa",
            "ext-value"
         ]
      },
      "duration": {
         "enum": [
            "second",
            "minute",
            "hour",
            "day",
            "month",
            "quarter",
            "year"
         ]
      },
      "lang": {
         "enum": ["en", "jp"]
      },
      "purpose": {
         "enum": ["traceback", "mitigation", "reporting", "watch", "other",
            "ext-value"
         ]
      },
      "restriction": {
         "enum": ["public", "partner", "need-to-know", "private",
            "default",
            "status": {
               "enum": ["new", "ext-value"
            ]
         ]
      },
   }
}
```

JSON schema is not yet published as an RFC. Meanwhile, CDDL is progressing rapidly.
Moving forward

1. There were several data types that were not defined in the version 03 draft. The 04 version will cope with that. (The github version already has addressed it.)

2. Please give us any feedback on the latest version.

3. How to reference JSON schema specification?