Transformation between XML and CBOR for network load reduction

Thomas Rix, Kai-Oliver Detken,
Marcel Jahnke

DECOIT GmbH
Fahrenheitstraße 9
D-28359 Bremen
https://www.decoit.de
info@decoit.de

Outline

1. Motivation
2. Transformation from XML to CBOR
3. Compression through usage of a dictionary
4. Real-life example: TCG IF-MAP

DECOIT®

Motivation

- XML is widely used for communication between software components
- Advantages over other formats, such as schema support
- Plain-text data formats take up high amounts of bandwith
  - Especially XML is a very wordy format
- Due to this not suitable for large amounts of individual packages
  - E.g. in SIEM systems
- CBOR allows concise representation of data and structuring data with arrays and maps (similar to JSON)

DECOIT®

From XML to CBOR

- Challenges:
  1. Preserve the sequence of elements
  2. Namespaces
  3. Attributes of elements
- All of these are unknown to JSON-like data formats
- Goal: Preserve all information stored in an XML document while keeping memory usage as low as possible

From XML to CBOR

- Challenge (1): Preserve the sequence of elements
- Solution: Use arrays as surrounding structures
  - Arrays have a fixed sequence for all contained elements
  - Only datatype in CBOR to ensure this

```
[
   <element>,
   <element>,
   <element>,
   ...
]
```

DECOIT®

From XML to CBOR

- Challenge (2): Namespaces
- Challenge (3): Attributes of elements

- Solutions: Use additional arrays to represent XML elements
  - A map would be the first guess, but requires additional and special keys for element name and value/nested elements
  - Thus adding payload not existent in the original document

DECOIT®

From CBOR to XML

- Representation of an XML element as array elements
  - Four elements
    1. Namespace (string)
    2. Element name (string)
    3. Attributes (array)
    4. Value (as required) / nested elements (array) / null
  - Each attribute consists of two array elements: key (string) and value (as required)
  - Multiple XML elements may be chained inside the same CBOR array
    - E.g. an array of eight elements for two XML elements

From CBOR to XML

XML

```
<element attribute="attr-value"
         xmlns="some-namespace">
    <nested-element attribute="attr-value"
         xmlns="some-namespace"/>
</element>
```

CBOR

```
[
   "some-namespace",
   "element",
   [
      "attribute",
      "attribute-value"
   ],
   [
      "some-namespace",
      "nested-element",
      [
         "attribute",
         "attribute-value"
      ],
      null
   ]
]
```

DECOIT®

Dictionary for compression

- Idea: Replace memory-intensive datatypes, such as strings, with more concise datatypes
  - CBOR integers can store values up to 23 in a single byte (including type information), making them a perfect candidate for a substitute datatype
- Goal: Define a dictionary with a simple plain-text file
  - Hierarchical structure to make 23 substitutes enough for large documents
  - Substitution for every possible static part of an XML document
    - namespace, element name, attribute name, enum values

Dictionary for compression

XML

```
<element attribute="attr-value"
         xmlns="some-namespace">
    <nested-element attribute="attr-value"
         xmlns="some-namespace"/>
</element>
```

Dictionary

```
n'some-namespace'[uint(0)] {
    t'element'[uint(0)] {
        a'attribute'[uint(0)]
        t'nested-element'[uint(0)] {
            a'attribute'[uint(0)]
        }
    }
}
```

Dictionary for compression

XML

```
<element attribute="attr-value"
         xmlns="some-namespace>
    <nested-element attribute="attr-value"
         xmlns="some-namespace"/>
</element>
```

CBOR

```
[
  "some-namespace",
  "element",
  [
    "attribute",
    "attribute-value"
  ],
  [
    "some-namespace",
    "nested-element",
    [
      "attribute",
      "attribute-value"
    ],
    null
  ]
]
```

CBOR compressed

```
[
  0,
  0,
  [
    0,
    "attribute-value"
  ],
  [
    0,
    0,
    [
      0,
      "attribute-value"
    ],
    null
  ]
]
```

Example: TCG IF-MAP
Publish-Delete Request

XML

- Plain: 358 Bytes
- GZIP: 251 Bytes

CBOR

- Plain: 256 Bytes
- GZIP: 167 Bytes
- Dict: 44 Bytes
- Dict+GZIP: 63 Bytes

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <ifmap:publish session-id="111"
          xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2">
      <delete>
        <ip-address type="IPv4" value="192.0.2.11"/>
        <mac-address value="00:11:22:33:44:55"/>
      </delete>
    </ifmap:publish>
  </soap:Body>
</soap:Envelope>
```

Example: TCG IF-MAP
Search Request

XML

- Plain: 506 Bytes
- GZIP: 331 Bytes

CBOR

- Plain: 365 Bytes
- GZIP: 255 Bytes
- Dict: 168 Bytes
- Dict+GZIP: 141 Bytes

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <ifmap:search
          match-links="meta:access-request-ip or meta:ip-mac or meta:access-request-mac"
          max-depth="3"
          result-filter="meta:capability or meta:device-attribute or meta:roles"
          session-id="111"
          terminal-identifier-type="identity,device"
          xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2">
      <ip-address type="IPv4" value="192.0.2.11"/>
    </ifmap:search>
  </soap:Body>
</soap:Envelope>
```

Thank you for listening!

DECOIT GmbH
Fahrenheitstraße 9
D-28359 Bremen

https://www.decoit.de
info@decoit.de