Subject Identifiers for Security Event Tokens

Annabelle Backman
IETF 103 – November 2018
Sub: Suboptimal for Some Scenarios

- Disambiguate multiple identifier types
  - Email
  - Phone #
  - OIDC subject ID
  - Token hashes

- Complex identifiers
  - OIDC issuer and subject
  - Token hash, key description, and algorithm
Subject Identifier Type

• "light-weight schema that describes a set of claims that uniquely identifies a subject."

• Name
  • email, phone, iss_sub

• Description of type of entity represented (e.g. user account associated with email)

• Supported claims
  • { email }, { phone }, { iss, sub }

• IANA Registry: “Security Event Subject Identifier Types”
Subject Identifier

• JSON object

• Type name in **subject_type** property

• Claims according to type definition
RISC Example: account_disabled

{  
  "iss": "https://risc.example.com/",
  "events": {
    "https://schemas.openid.net/secevent/risc/event-type/account-disabled": {
      "subject": {
        "subject_type": "iss-sub",
        "iss": "https://idp.example.com/",
        "sub": "7375626A656374",
      },
      "reason": "hijacking",
    }
  }
}
Current Status

• 02 draft published 2018-10-23

• Applications:
  • OIDF RISC

• Implementations:
  • Google: In progress
  • Amazon: In progress
• Added subject identifier type semantics.
## Subject Type Semantics

<table>
<thead>
<tr>
<th>Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>email</td>
<td>a user account associated with an email address</td>
</tr>
<tr>
<td>phone</td>
<td>a user account associated with a telephone number</td>
</tr>
<tr>
<td>iss-sub</td>
<td>an account identified by a pair of &quot;iss&quot; and &quot;sub&quot; claims, as defined by [JWT]</td>
</tr>
</tbody>
</table>
## Proposed Semantics

<table>
<thead>
<tr>
<th>Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>email</td>
<td>subject that can be communicated with via the specified email address</td>
</tr>
<tr>
<td>phone</td>
<td>subject that can be communicated with via the specified phone number</td>
</tr>
<tr>
<td>iss-sub</td>
<td>subject of a previously issued ID Token</td>
</tr>
</tbody>
</table>
Supporting Multiple Identifiers

• **Current solution: id-token-claims**
  • Supports combining any of iss-sub, email, and phone together.
  • Use case: event issuer does not know which identifier(s) event recipient uses.

```json
{
    "subject_type": "id-token-claims",
    "iss": "https://idp.example.com/",
    "sub": "7375626A656374",
    "email": "user@example.com",
    "phone_number": "+12065550100",
}
```
Problems with id-token-claims

- Confusing semantics.
- Duplicates other types.
- Other use cases:
  - Multiple email addresses, phone numbers?
  - Other identifier types?
Proposal: aliases

• List of subject identifiers with arbitrary types.
  • All MUST identify the same subject.
  • MAY contain multiple subject identifiers of the same type.

• Advantages:
  • Flexible.
  • Clear semantics.
  • Reuses other types.
  • Automatically supports new types.
  • Only affects those who need to use it.
Proposal: aliases (cont.)

{
    "subject_type": "aliases",
    "aliases": [
        {
            "subject_type": "iss-sub",
            "iss": "https://idp.example.com/",
            "sub": "7375626A656374",
        },
        {
            "subject_type": "email",
            "email": "user@example.com",
        }
    ],
}

Email Canonicalization

• Current draft is silent on the matter.

• Explicitly state that email is not canonicalized?

• Canonicalize?
  • How?