Security Events Working Group
Internet-Draft
Intended status: Standards Track
Expires: 26 December 2023

A. Backman, Ed.
Amazon
M. Scurtescu
Coinbase
P. Jain
Fastly

24 June 2023

Subject Identifiers for Security Event Tokens
draft-ietf-secevent-subject-identifiers-18

Abstract

Security events communicated within Security Event Tokens may support a variety of identifiers to identify subjects related to the event. This specification formalizes the notion of subject identifiers as structured information that describe a subject, and named formats that define the syntax and semantics for encoding subject identifiers as JSON objects. It also defines a registry for defining and allocating names for such formats, as well as the "sub_id" JSON Web Token (JWT) claim.

Table of Contents

1. Introduction

As described in Section 1.2 of SET [RFC8417], subjects related to security events may take a variety of forms, including but not limited to a JWT [RFC7519] principal, an IP address, a URL, etc. Different types of subjects may need to be identified in different ways (e.g., a user might be identified by an email address or a phone number or an account number). Furthermore, even in the case where the type of the subject is known, there may be multiple ways by which a given subject may be identified. For example, an account may be identified by an opaque identifier, an email address, a phone number, a JWT "iss" claim and "sub" claim, etc., depending on the nature and needs of the transmitter and receiver. Even within the context of a given transmitter and receiver relationship, it may be appropriate to identify different accounts in different ways, for example if some accounts only have email addresses associated with them while others only have phone numbers. Therefore it can be necessary to indicate within a SET the mechanism by which a subject is being identified.

To address this problem, this specification defines Subject Identifiers - JSON [RFC8259] objects containing information identifying a subject - and Identifier Formats - named sets of rules describing how to encode different kinds of subject identifying information (e.g., an email address, or an issuer and subject pair) as a Subject Identifier.

Below is a non-normative example of a Subject Identifier that identifies a subject by email address, using the Email Identifier Format.

```
{
  "format": "email",
  "email": "user@example.com"
}
```

Figure 1: Example: Subject Identifier using the Email Identifier Format

Subject Identifiers are intended to be a general-purpose mechanism for identifying subjects within JSON objects and their usage need not be limited to SETs. Below is a non-normative example of a JWT that uses a Subject Identifier in the "sub_id" claim (defined in this specification) to identify the JWT Subject.

```
{
  "iss": "issuer.example.com",
  "sub_id": {
    "format": "phone_number",
    "phone_number": "+12065550100"
  }
}
```

Figure 2: Example: JWT using a Subject Identifier with the "sub_id" claim

Usage of Subject Identifiers also need not be limited to identifying JWT Subjects. They are intended as a general-purpose means of expressing identifying information in an unambiguous manner. Below is a non-normative example of a SET containing a hypothetical security event describing the interception of a message, using Subject Identifiers to identify the sender, intended recipient, and interceptor.

```
{
  "iss": "issuer.example.com",
  "iat": 1508184845,
  "aud": "aud.example.com",
  "events": {
    "https://secevent.example.com/events/message-interception": {
      "from": {
        "format": "email",
        "email": "alice@example.com"
      },
      "to": {
        "format": "email",
        "email": "bob@example.com"
      },
      "interceptor": {
        "format": "email",
        "email": "eve@example.com"
      }
    }
  }
}
```

Figure 3: Example: SET with an event payload containing multiple Subject Identifiers

2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8417].

2.1. Definitions

This specification utilizes terminology defined in [RFC8259] and [RFC8417].

Within this specification, the terms "Subject" and "subject" refer generically to anything being identified via one or more pieces of information. The term "JWT Subject" refers specifically to the subject of a JWT (i.e., the subject that the JWT asserts claims about).

3. Subject Identifiers

A Subject Identifier is a JSON [RFC8259] object whose contents may be used to identify a subject within some context. An Identifier Format is a named definition of a set of information that may be used to identify a subject, and the rules for encoding that information as a Subject Identifier; they define the syntax and semantics of Subject Identifiers. A Subject Identifier MUST conform to a specific Identifier Format, and MUST contain a "format" member whose value is the name of that Identifier Format.

Every Identifier Format MUST have a unique name registered in the IANA "Security Event Identifier Formats" registry established by Section 8.1, or a Collision-Resistant Name as defined in [RFC7519]. Identifier Formats that are expected to be used broadly by a variety of parties SHOULD be registered in the "Security Event Identifier Formats" registry.

An Identifier Format MAY describe more members than are strictly necessary to identify a subject, and MAY describe conditions under which those members are required, optional, or prohibited. The "format" member is reserved for use as described in this specification; Identifier Formats MUST NOT declare any rules regarding the "format" member.

Every member within a Subject Identifier MUST match the rules specified for that member by this specification or by Subject Identifier's Identifier Format. A Subject Identifier MUST NOT contain any members prohibited or not described by its Identifier Format, and MUST contain all members required by its Identifier Format.

3.1. Identifier Formats versus Principal Types

Identifier Formats define how to encode identifying information for a subject. Unlike Principal Types, they do not define the type or nature of the subject itself. For example, while the "email" Identifier Format declares that the value of the "email" member is an email address, a subject in a Security Event that is identified by an "email" Subject Identifier could be an end user who controls that email address, the mailbox itself, or anything else that the transmitter and receiver both understand to be associated with that email address. Consequently Subject Identifiers remove ambiguity around how a subject is being identified, and how to parse an identifying structure, but do not remove ambiguity around how to resolve that identifier to a subject. For example, consider a directory management API that allows callers to identify users and groups through both opaque unique identifiers and email addresses. Such an API could use Subject Identifiers to disambiguate between which of these two types of identifiers is in use. However, the API would have to determine whether the subject is a user or group via some other means, such as by querying a database, interpreting other parameters in the request, or inferring the type from the API contract.

3.2. Identifier Format Definitions

The following Identifier Formats are registered in the IANA "Security Event Identifier Formats" registry established by Section 8.1.

Since the subject identifier format conveys semantic information, applications SHOULD choose the most specific possible format for the identifier in question. For example, an email address can be conveyed using a mailto: URI and the uri identifier format, but since the value is known to be an email address, the application should prefer to use the "email" identifier format instead.

3.2.1. Account Identifier Format

The Account Identifier Format identifies a subject using an account at a service provider, identified with an "acct" URI as defined in [RFC7565]. An account is an arrangement or agreement through which a user gets access to a service and gets a unique identity with the service provider. Subject Identifiers in this format MUST contain a "uri" member whose value is the "acct" URI for the subject. The "uri" member is REQUIRED and MUST NOT be null or empty. The Account Identifier Format is identified by a value of "account" in the "format" member.

Below is a non-normative example Subject Identifier for the Account Identifier Format:

```
{
  "format": "account",
  "uri": "acct:example.user@service.example.com"
}
```

Figure 4: Example: Subject Identifier for the Account Identifier Format

3.2.2. Email Identifier Format

The Email Identifier Format identifies a subject using an email address. Subject Identifiers in this format MUST contain an "email" member whose value is a string containing the email address of the subject, formatted as an "addr-spec" as defined in Section 3.4.1 of [RFC5322]. The "email" member is REQUIRED and MUST NOT be null or empty. The value of the "email" member MUST identify a mailbox to which email may be delivered, in accordance with [RFC5321]. The Email Identifier Format is identified by the name "email".

Below is a non-normative example Subject Identifier in the Email Identifier Format:

```
{
  "format": "email",
  "email": "user@example.com"
}
```

Figure 5: Example: Subject Identifier in the Email Identifier Format

3.2.2.1. Email Canonicalization

Many email providers will treat multiple email addresses as equivalent. While the domain portion of an [RFC5322] email address is consistently treated as case-insensitive per [RFC1034], most providers treat the local part of the email address as case-insensitive as well, and consider "user@example.com", "User@example.com", and "USER@example.com" as the same email address. Some providers also treat dots (".") as optional; for example, "user.name@example.com", "username@example.com", "u.s.e.r.name@example.com", and "u.s.e.r.n.a.m.e@example.com" might all be treated as equivalent. This has led users to view these strings as equivalent, driving service providers to implement proprietary email canonicalization algorithms to ensure that email addresses entered by users resolve to the same canonical string. Email canonicalization is not standardized, and there is no way for the event recipient to determine the mail providers canonicalization method. Therefore, the recipient SHOULD apply its own canonicalization algorithm to incoming events that reproduces the translation done by the local email system.

3.2.3. Issuer and Subject Identifier Format

The Issuer and Subject Identifier Format identifies a subject using a pair of "iss" and "sub" members, analogous to how subjects are identified using the "iss" and "sub" claims in OpenID Connect [OpenID.Core] ID Tokens. These members MUST follow the formats of the "iss" member and "sub" member defined by [RFC7519], respectively. Both the "iss" member and the "sub" member are REQUIRED and MUST NOT be null or empty. The Issuer and Subject Identifier Format is identified by the name "iss_sub".

Below is a non-normative example Subject Identifier in the Issuer and Subject Identifier Format:

```
{
  "format": "iss_sub",
  "iss": "https://issuer.example.com/",
  "sub": "145234573"
}
```

Figure 6: Example: Subject Identifier in the Issuer and Subject Identifier Format

3.2.4. Opaque Identifier Format

The Opaque Identifier Format describes a subject that is identified with a string with no semantics asserted beyond its usage as an identifier for the subject, such as a UUID or hash used as a surrogate identifier for a record in a database. Subject Identifiers in this format MUST contain an "id" member whose value is a JSON string containing the opaque string identifier for the subject. The "id" member is REQUIRED and MUST NOT be null or empty. The Opaque Identifier Format is identified by the name "opaque".

Below is a non-normative example Subject Identifier in the Opaque Identifier Format:

```
{
  "format": "opaque",
  "id": "11112222333344445555"
}
```

Figure 7: Example: Subject Identifier in the Opaque Identifier Format

3.2.5. Phone Number Identifier Format

The Phone Number Identifier Format identifies a subject using a telephone number. Subject Identifiers in this format MUST contain a "phone_number" member whose value is a string containing the full telephone number of the subject, including international dialing prefix, formatted according to E.164 [E164]. The "phone_number" member is REQUIRED and MUST NOT be null or empty. The Phone Number Identifier Format is identified by the name "phone_number".

Below is a non-normative example Subject Identifier in the Email Identifier Format:

```
{
  "format": "phone_number",
  "phone_number": "+12065550100"
}
```

Figure 8: Example: Subject Identifier in the Phone Number Identifier Format

3.2.6. Decentralized Identifier (DID) Format

The Decentralized Identifier Format identifies a subject using a Decentralized Identifier (DID) URL as defined in [DID]. Subject Identifiers in this format MUST contain a "URL" member whose value is a DID URL for the DID Subject being identified. The value of the "url" member MUST be a valid DID URL and MAY be a bare DID. The "url" member is REQUIRED and MUST NOT be null or empty. The Decentralized Identifier Format is identified by the name "did".

Below are non-normative example Subject Identifiers for the Decentralized Identifier Format:

```
{
  "format": "did",
  "url": "did:example:123456"
}
```

Figure 9: Example: Subject Identifier for the Decentralized Identifier Format, identifying a subject with a bare DID

```
{
  "format": "did",
  "url": "did:example:123456/did/url/path?versionId=1"
}
```

Figure 10: Example: Subject Identifier for the Decentralized Identifier Format, identifying a subject with a DID URL with non-empty path and query components

3.2.7. Uniform Resource Identifier (URI) Format

The Uniform Resource Identifier (URI) Format identifies a subject using a URI as defined in [RFC3986]. This identifier format makes no assumptions or guarantees with regard to the content, scheme, or reachability of the URI within the field. Subject Identifiers in this format MUST contain a "uri" member whose value is a URI for the subject being identified. The "uri" member is REQUIRED and MUST NOT be null or empty. The URI format is identified by the name "uri".

Below are non-normative example Subject Identifiers for the URI format:

```
{
  "format": "uri",
  "uri": "https://user.example.com/"
}
```

Figure 11: Example: Subject Identifier for the URI Format, identifying a subject with a website URI

```
{
  "format": "uri",
  "uri": "urn:uuid:4e851e98-83c4-4743-a5da-150ecb53042f"
}
```

Figure 12: Example: Subject Identifier for the URI Format, identifying a subject with a random URN

3.2.8. Aliases Identifier Format

The Aliases Identifier Format describes a subject that is identified with a list of different Subject Identifiers. It is intended for use when a variety of identifiers have been shared with the party that will be interpreting the Subject Identifier, and it is unknown which of those identifiers they will recognize or support. Subject Identifiers in this format MUST contain an "identifiers" member whose value is a JSON array containing one or more Subject Identifiers. Each Subject Identifier in the array MUST identify the same entity. The "identifiers" member is REQUIRED and MUST NOT be null or empty. It MAY contain multiple instances of the same Identifier Format (e.g., multiple Email Subject Identifiers), but SHOULD NOT contain exact duplicates. This format is identified by the name "aliases".

"aliases" Subject Identifiers MUST NOT be nested; i.e., the "identifiers" member of an "aliases" Subject Identifier MUST NOT contain a Subject Identifier in the "aliases" format.

Below is a non-normative example Subject Identifier in the Aliases Identifier Format:

```
{
  "format": "aliases",
  "identifiers": [
    {
      "format": "email",
      "email": "user@example.com"
    },
    {
      "format": "phone_number",
      "phone_number": "+12065550100"
    },
    {
      "format": "email",
      "email": "user+qualifier@example.com"
    }
  ]
}
```

Figure 13: Example: Subject Identifier in the Aliases Identifier Format

4. Subject Identifiers in JWTs

4.1. sub_id Claim

The "sub" JWT Claim is defined in Section 4.1.2 of [RFC7519] as containing a string value, and therefore cannot contain a Subject Identifier (which is a JSON object) as its value. This document defines the "sub_id" JWT Claim, in accordance with Section 4.2 of [RFC7519], as a common claim that identifies the JWT Subject using a Subject Identifier. When present, the value of this claim MUST be a Subject Identifier that identifies the subject of the JWT. The "sub_id" claim MAY be included in a JWT, whether or not the "sub" claim is present. When both the "sub" and "sub_id" claims are present in a JWT, they MUST identify the same subject, as a JWT has one and only one JWT Subject.

When processing a JWT with both "sub" and "sub_id" claims, implementations MUST NOT rely on both claims to determine the JWT Subject. An implementation MAY attempt to determine the JWT Subject from one claim and fall back to using the other if it determines it does not understand the format of the first claim. For example, an implementation may attempt to use "sub_id", and fall back to using "sub" upon finding that "sub_id" contains a Subject Identifier whose format is not recognized by the implementation.

Below are non-normative examples of JWTs containing the "sub_id" claim:

```
{
  "iss": "issuer.example.com",
  "sub_id": {
    "format": "email",
    "email": "user@example.com"
  }
}
```

Figure 14: Example: JWT containing a "sub_id" claim and no "sub" claim

```
{
  "iss": "issuer.example.com",
  "sub": "user@example.com",
  "sub_id": {
    "format": "email",
    "email": "user@example.com"
  }
}
```

Figure 15: Example: JWT where both the "sub" and "sub_id" claims identify the JWT Subject using the same identifier

```
{
  "iss": "issuer.example.com",
  "sub": "liz@example.com",
  "sub_id": {
    "format": "email",
    "email": "elizabeth@example.com"
  }
}
```

Figure 16: Example: JWT where both the "sub" and "sub_id" claims identify the JWT Subject using different values of the same identifier type

```
{
  "iss": "issuer.example.com",
  "sub": "user@example.com",
  "sub_id": {
    "format": "account",
    "uri": "acct:example.user@service.example.com"
  }
}
```

Figure 17: Example: JWT where the "sub" and "sub_id" claims identify the JWT Subject via different types of identifiers

4.2. sub_id and iss_sub Subject Identifiers

The "sub_id" claim MAY contain an "iss_sub" Subject Identifier. In this case, the JWT's "iss" claim and the Subject Identifier's "iss" member MAY be different. For example, in OpenID Connect [OpenID.Core] client may construct such a JWT when sending JWTs back to its OpenID Connect Identity Provider, in order to identify the JWT Subject using an identifier known to be understood by both parties. Similarly, the JWT's "sub" claim and the Subject Identifier's "sub" member MAY be different. For example, this may be used by an OpenID Connect client to communicate the JWT Subject's local identifier at the client back to its Identity Provider.

Below are non-normative examples of a JWT where the "iss" claim and "iss" member within the "sub_id" claim are the same, and a JWT where they are different.

```
{
  "iss": "issuer.example.com",
  "sub_id": {
    "format": "iss_sub",
    "iss": "issuer.example.com",
    "sub": "example_user"
  }
}
```

Figure 18: Example: JWT with an "iss_sub" Subject Identifier where JWT issuer and JWT Subject issuer are the same

```
{
  "iss": "client.example.com",
  "sub_id": {
    "format": "iss_sub",
    "iss": "issuer.example.com",
    "sub": "example_user"
  }
}
```

Figure 19: Example: JWT with an "iss_sub" Subject Identifier where the JWT issuer and JWT Subject issuer are different

```
{
  "iss": "client.example.com",
  "sub": "client_user",
  "sub_id": {
    "format": "iss_sub",
    "iss": "issuer.example.com",
    "sub": "example_user"
  }
}
```

Figure 20: Example: JWT with an "iss_sub" Subject Identifier where the JWT "iss" and "sub" claims differ from the JWT Subject's "iss" and "sub" members

5. Considerations for Specifications that Define Identifier Formats

Identifier Format definitions MUST NOT make assertions or declarations regarding the subject being identified by the Subject Identifier (e.g., an Identifier Format cannot be defined as specifically identifying human end users), as such statements are outside the scope of Identifier Formats and Subject Identifiers, and expanding that scope for some Identifier Formats but not others would harm interoperability, as applications that depend on this expanded scope to disambiguate the subject type would be unable to use Identifier Formats that do not provide such rules.

6. Privacy Considerations

6.1. Identifier Correlation

The act of presenting two or more identifiers for a single subject together (e.g., within an "aliases" Subject Identifier, or via the "sub" and "sub_id" JWT claims) may communicate more information about the subject than was intended. For example, the entity to which the identifiers are presented now knows that both identifiers relate to the same subject, and may be able to correlate additional data based on that. When transmitting Subject Identifiers, the transmitter SHOULD take care that they are only transmitting multiple identifiers together when it is known that the recipient already knows that the identifiers are related (e.g., because they were previously sent to the recipient as claims in an OpenID Connect ID Token), or when correlation is essential to the use case. Implementers must consider such risks, and specifications that use subject identifiers must provide appropriate privacy considerations of their own.

The considerations described in Section 6 of [RFC8417] also apply when Subject Identifiers are used within SETs. The considerations described in Section 12 of [RFC7519] also apply when Subject Identifiers are used within JWTs.

7. Security Considerations

This specification does not define any mechanism for ensuring the confidentiality or integrity of a Subject Identifier. Where such properties are required, implementations MUST use mechanisms provided by the containing format (e.g., integrity protecting SETs or JWTs using JWS [RFC7515]), or at the transport layer or other layer in the application stack (e.g., using TLS [RFC8446]).

Further considerations regarding confidentiality and integrity of SETs can be found in Section 5.1 of [RFC8417].

8. IANA Considerations

8.1. Security Event Identifier Formats Registry

This document defines Identifier Formats, for which IANA is asked to create and maintain a new registry titled "Security Event Identifier Formats". Initial values for the Security Event Identifier Formats registry are given in Section 3. Future assignments are to be made through the Specification Required registration policy [BCP26] and shall follow the template presented in Section 8.1.2.

It is suggested that multiple Designated Experts be appointed who are able to represent the perspectives of different applications using this specification, in order to enable broadly informed review of registration decisions.

8.1.1. Registry Location

(This section to be removed by the RFC Editor before publication as an RFC.)

The authors recommend that the Identifier Formats registry be located at https://www.iana.org/assignments/secevent/.

8.1.2. Registration Template

Format Name
: The name of the Identifier Format, as described in Section 3. The name MUST be an ASCII string consisting only of lower-case characters ("a" - "z"), digits ("0" - "9"), underscores ("_"), and hyphens ("-"), and SHOULD NOT exceed 20 characters in length.

Format Description
A brief description of the Identifier Format.

Change Controller
For formats defined in documents published by the IETF or its working groups, list "IETF". For all other formats, list the name of the party responsible for the registration. Contact information such as mailing address, email address, or phone number must also be provided.

Defining Document(s)
A reference to the document or documents that define the Identifier Format. The reference document(s) MUST specify the name, format,and meaning of each member that may occur within a Subject Identifier of the defined format, as well as whether each member is optional, required or conditional, and the circumstances under which these optional or conditional fields would be used. URIs that can be used to retrieve copies of each document SHOULD be included.

8.1.3. Initial Registry Contents

8.1.3.1. Account Identifier Format

* Format Name: "account"
* Format Description: Subject identifier based on acct URI.
* Change Controller: IETF
* Defining Document(s): Section 3 of this document.

8.1.3.2. Email Identifier Format

* Format Name: email
* Format Description: Subject identifier based on email address.
* Change Controller: IETF
* Defining Document(s): Section 3 of this document.

8.1.3.3. Issuer and Subject Identifier Format

* Format Name: "iss_sub"
* Format Description: Subject identifier based on an issuer and subject.

* Change Controller: IETF

* Defining Document(s): Section 3 of this document.

8.1.3.4. Opaque Identifier Format

* Format Name: "opaque"

* Format Description: Subject identifier based on an opaque string.

* Change Controller: IETF

* Defining Document(s): Section 3 of this document.

8.1.3.5. Phone Number Identifier Format

* Format Name: "phone_number"

* Format Description: Subject identifier based on an phone number.

* Change Controller: IETF

* Defining Document(s): Section 3 of this document.

8.1.3.6. Decentralized Identifier Format

* Format Name: "did"

* Format Description: Subject identifier based on a decentralized identifier (DID).

* Change Controller: IETF

* Defining Document(s): Section 3 of this document.

8.1.3.7. Uniform Resource Identifier Format

* Format Name: "uri"

* Format Description: Subject identifier based on a uniform resource identifier (URI).

* Change Controller: IETF

* Defining Document(s): Section 3 of this document.

8.1.3.8. Aliases Identifier Format

* Format Name: "aliases"

* Format Description: Subject identifier that groups together multiple different subject identifiers for the same subject.

* Change Controller: IETF

* Defining Document(s): Section 3 of this document.

8.1.4. Guidance for Expert Reviewers

The Expert Reviewer is expected to review the documentation referenced in a registration request to verify its completeness. The Expert Reviewer must base their decision to accept or reject the request on a fair and impartial assessment of the request. If the Expert Reviewer has a conflict of interest, such as being an author of a defining document referenced by the request, they must recuse themselves from the approval process for that request.

Identifier Formats need not be generally applicable and may be highly specific to a particular domain; it is expected that formats may be registered for niche or industry-specific use cases. The Expert Reviewer should focus on whether the format is thoroughly documented, and whether its registration will promote or harm interoperability. In most cases, the Expert Reviewer should not approve a request if the registration would contribute to confusion, or amount to a synonym for an existing format.

8.2. JSON Web Token Claims Registration

This document defines the sub_id JWT Claim, which IANA is asked to register in the "JSON Web Token Claims" registry IANA JSON Web Token Claims Registry [IANA.JWT.Claims] established by [RFC7519].

8.2.1. Registry Contents

* Claim Name: "sub_id"

* Claim Description: Subject Identifier

* Change Controller: IETF

* Specification Document(s): Section 4.1 of this document.

9. References

9.1. Normative References

[BCP26] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[DID] World Wide Web Consortium (W3C), "Decentralized Identifiers (DIDs) v1.0", 2021, <https://www.w3.org/TR/did-core/>.

[E164] International Telecommunication Union, "The international public telecommunication numbering plan", 2010, <https://www.itu.int/rec/T-REC-E.164-201011-I/en>.

[IANA.JWT.Claims] IANA, "JSON Web Token Claims", n.d., <https://www.iana.org/assignments/jwt>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <https://www.rfc-editor.org/info/rfc5321>.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <https://www.rfc-editor.org/info/rfc5322>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[RFC7565] Saint-Andre, P., "The 'acct' URI Scheme", RFC 7565, DOI 10.17487/RFC7565, May 2015, <https://www.rfc-editor.org/info/rfc7565>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[RFC8417] Hunt, P., Ed., Jones, M., Denniss, W., and M. Ansari, "Security Event Token (SET)", RFC 8417, DOI 10.17487/RFC8417, July 2018, <https://www.rfc-editor.org/info/rfc8417>.

9.2. Informative References

[OpenID.Core] Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., and C. Mortimore, "OpenID Connect Core 1.0", November 2014, <https://openid.net/specs/openid-connect-core-1_0.html>.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

Acknowledgements

The authors would like to thank the members of the IETF Security Events working group, as well as those of the OpenID Shared Signals and Events Working Group, whose work provided the original basis for this document. We would also like to acknowledge Aaron Parecki, Denis Pinkas, Justin Richer, Mike Jones and other members of the working group for reviewing this document.

Change Log

(This section to be removed by the RFC Editor before publication as an RFC.)

Draft 00 - AB - First draft

Draft 01 - AB:

* Added reference to RFC 5322 for format of email claim.

* Renamed iss_sub type to iss-sub.

* Renamed id_token_claims type to id-token-claims.

* Added text specifying the nature of the subjects described by each type.

Draft 02 - AB:

* Corrected format of phone numbers in examples.

* Updated author info.

Draft 03 - AB:

* Added account type for acct URIs.

* Replaced id-token-claims type with aliases type.

* Added email canonicalization guidance.

* Updated semantics for email, phone, and iss-sub types.

Draft 04 - AB:

* Added sub_id JWT Claim definition, guidance, examples.

* Added text prohibiting aliases nesting.

* Added privacy considerations for identifier correlation.

Draft 05 - AB:

* Renamed the phone type to phone-number and its phone claim to phone_number.

Draft 06 - AB:

* Replaced usage of the word "claim" to describe members of a Subject Identifier with the word "member", in accordance with terminology in RFC8259.

* Renamed the phone-number type to phone_number and iss-sub to iss_sub.

* Added normative requirements limiting the use of both sub and sub_id claims together when processing a JWT.

* Clarified that identifier correlation may be acceptable when it is a core part of the use case.

* Replaced references to OIDF with IETF in IANA Considerations.

* Recommended the appointment of multiple Designated Experts, and a location for the Subject Identifier Types registry.

* Added "_" to list of allowed characters in the Type Name for Subject Identifier Types.

* Clarified that Subject Identifiers don't provide confidentiality or integrity protection.

* Added references to SET, JWT privacy and security considerations.

* Added section describing the difference between subject identifier type and principal type that hopefully clarifies things and doesn't just muddy the water further.

Draft 07 - AB:

* Emphasized that the spec is about identifiers, not the things they identify:

  - Renamed "Subject Identifier Type" to "Identifier Format".

  - Renamed subject_type to format.

  - Renamed "Security Event Subject Identifier Type Registry" to "Security Event Identifier Format Registry".

  - Added new section with guidance for specs defining Identifier Formats, with normative prohibition on formats that describe the subject itself, rather than the identifier.

* Clarified the meaning of "subject":

  - Defined "subject" as applying generically and "JWT Subject" as applying specifically to the subject of a JWT.

  - Replaced most instances of the word "principal" with "subject".

* Added opaque Identifier Format

Draft 08 - JR, AB:

* Added did Identifier Format

* Alphabetized identifier format definitions

* Replaced "type" with "format" in places that had been missed in the -07 change. (mostly IANA Considerations)

* Miscellaneous editorial fixes

Draft 09 - AB:

* Miscellaneous editorial fixes

Draft 10 - PJ:

* Added author

* Editorial nits

Draft 11 - PJ:

* Miscellaneous editorial fixes

* Moved aliases to the last in identifier format definitions

* Acknowledged individual reviewers

Draft 12 - PJ:

* Restore the DID format that was removed in -11

* Added a generic "URI" format

* Normative advice on choosing the format

Draft 13 - PJ:

* Editorial nits found during AD review

Draft 14 - PJ:

* Fix IANA issues found during AD review

Draft 15 - PJ:

* Fix issues found during review

Draft 16 - PJ:

* Change controller updated to IETF

Draft 17 -PJ:

* Fixed nits identified during IESG reviews

Draft 18 -PJ:

* Fixed issues identified during IESG reviews

Authors' Addresses

Annabelle Backman (editor)
Amazon
Email: richanna@amazon.com

Marius Scurtescu
Coinbase
Email: marius.scurtescu@coinbase.com

Prachi Jain
Fastly
Email: prachi.jain1288@gmail.com