Security Events Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 27, 2017

P. Hunt, Ed.
Oracle
W. Denniss
Google
M. Ansari
Cisco
M. Jones
Microsoft

November 23, 2016

Security Event Token (SET)
draft-hunt-idevent-token-07

Abstract

This specification defines the Security Event Token, which may be distributed via a protocol such as HTTP. The Security Event Token (SET) specification profiles the JSON Web Token (JWT) and may be optionally signed and/or encrypted. A SET describes a statement of fact that may be shared by an event publisher with event subscribers.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction and Overview

This specification defines an extensible Security Event Token (SET) format which may be exchanged using protocols such as HTTP. The specification builds on the JSON Web Token (JWT) format [RFC7519] in order to provide a self-contained token that can be optionally signed using JSON Web Signature (JWS) [RFC7515] and/or encrypted using JSON Web Encryption (JWE) [RFC7516].

For the purpose of this specification, an event is a statement of fact by a publisher (also known as the event issuer) that the state of a security subject (e.g., a web resource, token, IP address) it controls or is aware of, has changed in some way (explicitly or implicitly). A security subject may be permanent (e.g., a user account) or temporary (e.g., a login session) in nature. A state change may include direct changes of entity state, implicit changes to state or other higher-level security statements such as:

- The creation, modification, removal of a resource.

- The resetting or suspension of an account.

- The revocation of a security token prior to its expiry.

- The logout of a user session. Or,

- A cumulative conclusion such as to indicate that a user has taken over an email identifier that may have been used in the past by another user.

Based on some agreed upon criteria for an event feed, the publisher distributes events to the appropriate subscribers. While an event may be delivered via synchronous means (e.g., HTTP POST), the distribution of the event often happens asynchronously to the change of state which generated the security event. As an example, an OAuth2 Authorization Server [RFC6749], having received a token revocation request [RFC7009], may issue a token revocation event to downstream web resource providers. Having been informed of a token revocation, the OAuth2 web resource service provider may add the token identifier to its local revocation list assuming the token has not already expired.

A subscriber having received an event, validates and interprets the event and takes its own independent action, if any. For example, having been informed of a personal identifier now being associated with a different security subject (i.e., is being used by someone else), the subscriber may choose to ensure that the new user is not granted access to resources associated with the previous user. Or it may not have any relationship with the subject, and no action is taken.

While subscribers will often take actions upon receiving one or more events, events MUST NOT be assumed to be commands or requests. To do so requires complex bi-directional signals and error recovery mechanisms that fall outside the scope of this specification. The intent of this specification is to define a way of exchanging statements of fact that subscribers may interpret for their own purposes. Since events are typically historical statements by a publisher and are not commands, idempotency or lack thereof, does not apply.

Unless otherwise specified, this specification uses example events intended as non-normative examples showing how an event may be used. It is expected that other specifications will use this specification to define normative events.

This specification is scoped to security and identity related events. While security event tokens may be used for other purposes, the specification only considers security and privacy concerns relevant to identity and personal information.

1.1. 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 [RFC2119]. These keywords are capitalized when used to unambiguously specify requirements of the protocol or application features and behavior that affect the inter-operability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

For purposes of readability, examples are not URL encoded. Implementers MUST percent encode URLs as described in Section 2.1 of [RFC3986].

Throughout this document, all figures MAY contain spaces and extra line-wrapping for readability and space limitations. Similarly, some URIs contained within examples have been shortened for space and readability reasons.

1.2. Definitions

The following definitions are used with SETs:

Feed Publisher
The Feed Publisher creates SETs to be distributed to registered subscribers. In JWT terminology, the Feed Publisher is also known as the issuer ("iss").

Security Event Token (SET)
An SET is a JWT that is to be distributed to one or more registered subscribers. A SET MAY be signed or encrypted using JWS and/or JWE for authentication and confidentiality reasons.

Feed
A Feed is a logical grouping of SETs or a context under which SETs may be issued. A Subscriber registers with the Feed Publisher to subscribe to SETs associated with a Feed. How a Feed is defined or the method for subscription is out-of-scope of this specification.

Subscriber

A Subscriber registers to receive SETs from a Feed Publisher using a protocol such as HTTP. The method of registration and delivery is out-of-scope of this specification.

Security Subject
A Security Subject is the entity to which a SET refers. A Security Subject may be a principle (e.g., Section 4.1.2 [RFC7519]), a web resource, or other thing such as an IP address that a SET might reference.

2. The Security Event Token (SET)

A SET conveys a statement (in the form of a JWT [RFC7519]) about a single security event in relation to a Security Subject that may be of interest to a Subscriber or set of Subscribers receiving SETs from a Feed Publisher.

The schema and structure of a SET follows the JWT [RFC7519] specification. A SET has the following structure:

o An outer JSON structure that acts as the SET envelope. The envelope contains a set of name/value pairs called the JWT Claims Set, typically common to every SET or common to a number of different Security Events within a single profiling specification or a related series of specifications. Claims in the envelope SHOULD be registered in the JWT Token Claims Registry Section 10.1 [RFC7519] or be Public Claims or Private Claims as also defined in [RFC7519].

o Envelope claims that are profiled and defined in this specification are used to validate the SET and determine the event data included. The claim "events" identifies the type of security event and MAY also include event-specific data. While a SET contains a single event, it MAY have multiple extensions providing additional data about the same event. The primary event is typically the first value in the "events" object, while event extensions are the 2nd, 3rd, etc.

o Each JSON member of the "events" object is a name/value pair, whose value is a JSON object known as the event "payload". The payload object contains claims typically unique to the event's URI value and are not registered as JWT claims. These claims are defined by their associated event specification. An event with no payload claims SHALL be represented as the empty JSON object ("{}"). Event extensions can be used for many purposes. Some examples include but are not limited to:

- A categorization extension applied to multiple event types to provide classification information (e.g., threat type or level).

- Enhancement of an existing specifications the arise over time.

- Correlation extensions needed to link a potential series of events.

- Localized contextual extensions needed between a publisher and subscriber.

The following is a non-normative example showing the JWT Claims Set for a hypothetical SCIM password reset SET. This example is also one in which the issuer has provided an extension ("https://example.com/scim/event/passwordResetExt") that is used to convey additional information -- in this case, the current count of reset attempts:

```
{
  "jti": "3d0c3cf797584bd193bd0fb1bd4e7d30",
  "iat": 1458496025,
  "iss": "https://scim.example.com",
  "aud": [
    "https://jhub.example.com/Feeds/98d52461fa5bbc879593b7754",
    "https://jhub.example.com/Feeds/5d7604516b1d08641d7676ee7"
  ],
  "sub": "https://scim.example.com/Users/44f6142df96bd6ab61e7521d9",
  "events": {
    "urn:ietf:params:scim:event:passwordReset":
      { "id":"44f6142df96bd6ab61e7521d9"},
    "https://example.com/scim/event/passwordResetExt":
      { "resetAttempts":5}
  }
}
```

Figure 1: Example SCIM Password Reset Event

The event in the figure above expresses hypothetical password reset event for SCIM [RFC7644]. The JWT consists of:

- An "events" claim specifying the hypothetical SCIM URN ("urn:ietf:params:scim:event:passwordReset") for a password reset, and a custom extension, "https://example.com/scim/event/passwordResetExt", that is used to provide additional event information such as the current count of resets.

- An "iss" claim, denoting the event publisher.

- The "sub" claim specifies the SCIM resource URI that was affected.

- The "aud" claim specifies the intended audiences for the event. In practical terms, an audience MAY be the URI for an event feed that a client has subscribed to.

Additional extensions to an event may be added by adding more values to the "events" claims. For each event URI value specified, there is a corresponding JSON object that contains the claims associated with that event, if any. In this example, the SCIM event indicates that a password has been updated and the current password reset count is 5. Notice that the value for "resetAttempts" is actually part of its own JSON object associated with the extension URI.

Here is another example JWT Claims Set for a security event token, this one for a Logout Token:

```
{
   "iss": "https://server.example.com",
   "sub": "248289761001",
   "aud": "s6BhdRkqt3",
   "iat": 1471566154,
   "jti": "bWJq",
   "sid": "08a5019c-17e1-4977-8f42-65a12843ea02",
   "events": {
     "http://schemas.openid.net/event/backchannel-logout": {}
   }
}
```

Figure 2: Example OpenID Back-Channel Logout Event

Note that the above SET has an empty JSON object and uses the JWT registered claims "sub" and "sid" to identify the subject that was logged-out.

In the following example JWT Claims Set, a fictional medical service collects consent for medical actions and notifies other parties. The individual for whom consent is identified was originally authenticated via OpenID Connect. In this case, the issuer of the security event is an application rather than the OpenID provider:

```
{
  "jti": "fb4e75b5411e4e19b6c0fe87950f7749",

  "sub": "248289761001",
  "iat": 1458496025,
  "iss": "https://my.examplemed.com",
  "aud": [
    "https://rp.example.com"
  ],
  "events": {
    "https://openid.net/heart/consent.html":{
      "consentUri":[
        "https://terms.examplemed.com/labdisclosure.html#Agree"
      ]
    }
  }
}
```

Figure 3: Example Consent Event

In the above example "iss" and "sub" contained within the claim "https://openid.net/heart/consent", refer to the subject and issuer of the original OpendID Provider. They are distinct from the top level value of "iss" which always refers to the issuer of the event - a medical consent service that is a relying party to the OpenID Provider.

2.1. Core SET Claims

The following are claims that are based on [RFC7519] claim definitions and are profiled for use in an event token:

jti

As defined by Section 4.1.7 [RFC7519] contains a unique identifier for an event. The identifier SHOULD be unique within a particular event feed and MAY be used by clients to track whether a particular event has already been received. This claim is REQUIRED.

iss

A single valued String containing the URI of the service provider publishing the SET (the issuer). This claim is REQUIRED.

aud
A multi-valued String containing the URIs representing the audience of the event. Values are typically URLs of the feeds the event is associated with. When an event has multiple audiences that go to the same subscriber, the publisher is not obligated to deliver repeated events to the same subscriber. This claim is RECOMMENDED.

iat
As defined by Section 4.1.6 [RFC7519], a value containing a NumericDate, which represents when the event was issued. Unless otherwise specified, the value SHOULD be interpreted by the subscriber as equivalent to the actual time of the event. This claim is REQUIRED.

nbf
As defined by Section 4.1.5 [RFC7519], a value containing a NumericDate, which represents a future date when the event will occur. This claim is OPTIONAL.

sub As defined by Section 4.1.2 [RFC7519], a String or URI value representing the principal or the subject of the SET. This is usually the entity whose "state" was changed. For example, an IP Address was added to a black list. A URI representing a user resource that was modified. A token identifier for a revoked token. If used, the profile specification SHOULD define the content and format semantics for the value. This claim is OPTIONAL, as the principal for any given profile may already be identified without the inclusion of a subject claim.

exp As defined by [RFC7519], this claim is time on which the JWT MUST NOT be accepted for processing. In the context of a SET however, this notion does not apply since a SET reflects something that has already been processed and is historical in nature. While some specifications MAY have a need for this claim, its use in general cases is NOT RECOMMENDED.

The following are new claims defined by this specification:

events
A JSON object whose members are a set of JSON name/value pairs whose names are URIs representing the primary event (typically the first member) and event extensions being expressed. For each name present, the corresponding value SHALL be a JSON object. The JSON object MAY be an empty object ("{}"), or it MAY be a JSON object containing data as described by the profiling event specification.

txn

An OPTIONAL single-valued String value that represents a unique transaction identifier. In cases where multiple SETs are issued based on different event URIs, the transaction identifier MAY be used to correlate SETs to the same originating event or stateful change.

2.2. Security Event Token Construction

A SET is a JWT [RFC7519] that is constructed by building a JSON structure that constitutes an event object and which is then used as the body of a JWT.

While this specification uses JWT to convey a SET, implementers SHALL NOT use SETs to convey authentication or authorization assertions.

The following is an example JWT Claims Set for a security event token (which has been formatted for readability):

```
{
  "jti": "4d3559ec67504aaba65d40b0363faad8",
  "iat": 1458496404,
  "iss": "https://scim.example.com",
  "aud": [
   "https://scim.example.com/Feeds/98d52461fa5bbc879593b7754",
   "https://scim.example.com/Feeds/5d7604516b1d08641d7676ee7"
  ],

  "events": {
    "urn:ietf:params:scim:event:create": {
      "ref":
        "https://scim.example.com/Users/44f6142df96bd6ab61e7521d9",
      "attributes":["id", "name", "userName", "password", "emails"],
      "values": {
        "emails": [
         {"type": "work", "value": "jdoe@example.com"}
        ],
        "password": "not4u2no",
        "userName": "jdoe",
        "id": "44f6142df96bd6ab61e7521d9",
        "name": {
          "givenName": "John",
          "familyName": "Doe"
        }
      }
    }
  }
}
```

Figure 4: Example Event Claims

When transmitted, the above JSON body must be converted into a JWT as per [RFC7519]. In this example, because the event contains attribute values, the token MUST be encrypted per JWE (see [RFC7516]) before transmission.

The following is an example of a SCIM Event expressed as an unsecured JWT. The JWT header of:

```
{"alg":"none"}
```

Base64url encoding of the octets of the UTF-8 representation of the header yields:

```
eyJhbGciOiJub25lIn0
```

The example JSON Event Data is encoded as follows:

```
eyAgCiAgImp0aSI6ICI0ZDM1NTllYzY3NTA0YWFiYTY1ZDQwYjAzNjNmYWFkOCIsCiAg
ImlhdCI6IDE0NTg0OTY0MDQsCiAgImlzcyI6ICJodHRwczovL3NjaW0uZXhhbXBsZS5j
b20iLCAgCiAgImF1ZCI6IFsKICAgImh0dHBzOi8vc2NpbS5leGFtcGxlLmNvbS9GZWVk
cy85OGQ1MjQ2MWZhNWJiYzg3OTU5M2I3NzU0IiwKICAgImh0dHBzOi8vc2NpbS5leGFt
cGxlLmNvbS9GZWVkcy81ZDc2MDQ1MTZiMWQwODY0MWQ3Njc2ZWU3IgogIF0sICAKICAK
ICAiZXZlbnRzIjogewogICAgInVybjppZXRmOnBhcmFtczpzY2ltOmV2ZW50OmNyZWF0
ZSI6IHsKICAgICAgInJlZiI6CiAgICAgICAgImh0dHBzOi8vc2NpbS5leGFtcGxlLmNv
bS9Vc2Vycy80NGY2MTQyZGY5NmJkNmFiNjFlNzUyMWQ5IiwKICAgICAgImF0dHJpYnV0
ZXMiOlsiaWQiLCAibmFtZSIsICJ1c2VyTmFtZSIsICJwYXNzd29yZCIsICJlbWFpbHMi
XSwKICAgICAgInZhbHVlcyI6IHsKICAgICAgICAiZW1haWxzIjogWwogICAgICAgICB7
InR5cGUiOiAid29yayIsICJ2YWx1ZSI6ICJqZG9lQGV4YW1wbGUuY29tIn0KICAgICAg
ICBdLAogICAgICAgICJwYXNzd29yZCI6ICJub3Q0dTJubyIsCiAgICAgICAgInVzZXJO
YW1lIjogImpkb2UiLAogICAgICAgICJpZCI6ICI0NGY2MTQyZGY5NmJkNmFiNjFlNzUy
MWQ5IiwKICAgICAgICAibmFtZSI6IHsKICAgICAgICAgICJnaXZlbk5hbWUiOiAiSm9o
biIsCiAgICAgICAgICAiZmFtaWx5TmFtZSI6ICJEb2UiCiAgICAgICAgfQogICAgICB9
CiAgICB9CiAgfQp9
```

The encoded JWS signature is the empty string. Concatenating the parts yields:

```
eyJhbGciOiJub25lIn0
.
eyAgCiAgImp0aSI6ICI0ZDM1NTllYzY3NTA0YWFiYTY1ZDQwYjAzNjNmYWFkOCIsCiAg
ImlhdCI6IDE0NTg0OTY0MDQsCiAgImlzcyI6ICJodHRwczovL3NjaW0uZXhhbXBsZS5j
b20iLCAgCiAgImF1ZCI6IFsKICAgImh0dHBzOi8vc2NpbS5leGFtcGxlLmNvbS9GZWVk
cy85OGQ1MjQ2MWZhNWJiYzg3OTU5M2I3NzU0IiwKICAgImh0dHBzOi8vc2NpbS5leGFt
cGxlLmNvbS9GZWVkcy81ZDc2MDQ1MTZiMWQwODY0MWQ3Njc2ZWU3IgogIF0sICAKICAK
ICAiZXZlbnRzIjogewogICAgInVybjppZXRmOnBhcmFtczpzY2ltOmV2ZW50OmNyZWF0
ZSI6IHsKICAgICAgInJlZiI6CiAgICAgICAgImh0dHBzOi8vc2NpbS5leGFtcGxlLmNv
bS9Vc2Vycy80NGY2MTQyZGY5NmJkNmFiNjFlNzUyMWQ5IiwKICAgICAgImF0dHJpYnV0
ZXMiOlsiaWQiLCAibmFtZSIsICJ1c2VyTmFtZSIsICJwYXNzd29yZCIsICJlbWFpbHMi
XSwKICAgICAgInZhbHVlcyI6IHsKICAgICAgICAiZW1haWxzIjogWwogICAgICAgICB7
InR5cGUiOiAid29yayIsICJ2YWx1ZSI6ICJqZG9lQGV4YW1wbGUuY29tIn0KICAgICAg
ICBdLAogICAgICAgICJwYXNzd29yZCI6ICJub3Q0dTJubyIsCiAgICAgICAgInVzZXJO
YW1lIjogImpkb2UiLAogICAgICAgICJpZCI6ICI0NGY2MTQyZGY5NmJkNmFiNjFlNzUy
MWQ5IiwKICAgICAgICAibmFtZSI6IHsKICAgICAgICAgICJnaXZlbk5hbWUiOiAiSm9o
biIsCiAgICAgICAgICAiZmFtaWx5TmFtZSI6ICJEb2UiCiAgICAgICAgfQogICAgICB9
CiAgICB9CiAgfQp9
.
```

Figure 5: Example Unsecured Security Event Token

To create and or validate a signed or encrypted SET, follow the instructions in section 7 of [RFC7519].

3. Security Considerations

3.1. Confidentiality and Integrity

SETs may often contain sensitive information. Therefore, methods for distribution of events SHOULD require the use of a transport-layer security mechanism when distributing events. Parties MUST support TLS 1.2 [RFC5246] and MAY support additional transport-layer mechanisms meeting its security requirements. When using TLS, the client MUST perform a TLS/SSL server certificate check, per [RFC6125]. Implementation security considerations for TLS can be found in "Recommendations for Secure Use of TLS and DTLS" [RFC7525].

Security Events distributed through third-parties or that carry personally identifiable information, SHOULD be encrypted using JWE [RFC7516] or secured for confidentiality by other means.

Security Events distributed without authentication over the channel, such as via TLS ([RFC5246] and [RFC6125]), and/or OAuth2 [RFC6749], or Basic Authentication [RFC7617], MUST be signed using JWS [RFC7515] so that individual events MAY be authenticated and validated by the subscriber.

3.2. Delivery

This specification does not define a delivery mechanism by itself. In addition to confidentiality and integrity (discussed above), implementers and profile specifications MUST consider the consequences of delivery mechanisms that are not secure and/or not assured. For example, while a SET may be end-to-end secured using JWE, that alone will not guarantee that the correct subscribing party knows they should have received a particular SET.

3.3. Sequencing

As defined in this specification, there is no defined way to order multiple SETs in a sequence. Depending on the type and nature of SET event, order may or may not matter. For example, in provisioning, event order is critical -- an object could not be modified before it was created. In other SET types, such as a token revocation, the order of SETs for revoked tokens does not matter. If however, the event was described as a log-in or logged-out status for a user subject, then order becomes important.

Extension specifications and implementers SHOULD take caution when using timestamps such as "iat" to define order. Distributed systems will have some amount of clock-skew and thus time by itself will not guarantee order.

Specifications profiling SET SHOULD define a mechanism for detecting order or sequence of events. For example, the "txn" claim could contain an ordered value (e.g., a counter) that the publisher defines.

3.4. Timing Issues

When SETs are delivered asynchronously and/or out-of-band with respect to the original action that incurred the security event, it is important to consider that a SET might be delivered to a Subscriber in advance or well behind the process that caused the event. For example, a user having been required to logout and then log back in again, may cause a logout SET to be issued that may arrive at the same time as the user-agent accesses a web site having just logged-in. If timing is not handled properly, the effect would be to erroneously treat the new user session as logged out. Profiling specifications SHOULD be careful to anticipate timing and subject selection information. For example, it might be more appropriate to cancel a "session" rather than a "user". Alternatively, the specification could use timestamps that allows new sessions to be started immediately after a stated logout event time.

3.5. Distinguishing SETs from Access Tokens

Because [RFC7519] states that "all claims that are not understood by implementations MUST be ignored.", there is a consideration that a SET token might be confused as an access or authorization token in the case where a SET is mistakenly or intentionally intercepted and presented as an access token. To avoid this, it is recommended that implementers consider one or more of the following:

- Avoid use of the JWT claim "exp" within the envelope.

- Where possible, use a separate "aud" claim value to distinguish between the SET subscriber and the audience of an access token. For example, a Logout while intended for the same relying party could use a different audience to distinguish between normal access and logout notification.

- Modify access validation systems to check for the presence of the "events" claim as a means to detect security event tokens. This is particularly useful if the same endpoint may receive both types of tokens.

- Consider avoiding use of the "sub" claim at the top level.

4. Privacy Considerations

If a SET needs to be retained for audit purposes, JWS MAY be used to provide verification of its authenticity.

Event Publishers SHOULD attempt to specialize feeds so that the content is targeted to the specific business and protocol needs of subscribers.

When sharing personally identifiable information or information that is otherwise considered confidential to affected users, the publishers and subscribers MUST have the appropriate legal agreements and user consent in place.

The propagation of subject identifiers can be perceived as personally identifiable information. Where possible, publishers and subscribers should devise approaches that prevent propagation -- for example, the passing of a hash value that requires the subscriber to already know the subject.

5. IANA Considerations

5.1. JSON Web Token Claims Registration

This specification registers the "events" and "txn" claims in the IANA "JSON Web Token Claims" registry [IANA.JWT.Claims] established by [RFC7519].

5.1.1. Registry Contents

- Claim Name: "events"
- Claim Description: Security Event Object
- Change Controller: IESG
- Specification Document(s): Section 2 of [[this specification]]

- Claim Name: "txn"
- Claim Description: Transaction Identifier
- Change Controller: IESG
- Specification Document(s): Section 2 of [[this specification]]

6. References

6.1. Normative References

[IANA.JWT.Claims] IANA, "JSON Web Token Claims", <http://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, <http://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, <http://www.rfc-editor.org/info/rfc3986>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <http://www.rfc-editor.org/info/rfc6125>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <http://www.rfc-editor.org/info/rfc6749>.

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

[RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <http://www.rfc-editor.org/info/rfc7525>.

[RFC7617] Reschke, J., "The 'Basic' HTTP Authentication Scheme", RFC 7617, DOI 10.17487/RFC7617, September 2015, <http://www.rfc-editor.org/info/rfc7617>.

6.2. Informative References

[RFC7009] Lodderstedt, T., Ed., Dronia, S., and M. Scurtescu, "OAuth 2.0 Token Revocation", RFC 7009, DOI 10.17487/RFC7009, August 2013, <http://www.rfc-editor.org/info/rfc7009>.

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

[RFC7516] Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, DOI 10.17487/RFC7516, May 2015, <http://www.rfc-editor.org/info/rfc7516>.

[RFC7517] Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015, <http://www.rfc-editor.org/info/rfc7517>.

[RFC7644] Hunt, P., Ed., Grizzle, K., Ansari, M., Wahlstroem, E., and C. Mortimore, "System for Cross-domain Identity Management: Protocol", RFC 7644, DOI 10.17487/RFC7644, September 2015, <http://www.rfc-editor.org/info/rfc7644>.

Appendix A. Acknowledgments

The editors would like to thank the participants in the IETF id-event mailing list and related working groups for their support of this specification.

Appendix B. Change Log

Draft 01 - PH - Renamed eventUris to events

Draft 00 - PH - First Draft

Draft 01 - PH - Fixed some alignment issues with JWT. Remove event type attribute.

Draft 02 - PH - Renamed to Security Events, removed questions, clarified examples and intro text, and added security and privacy section.

Draft 03 - PH

General edit corrections from Sarah Squire
Changed "event" term to "SET"
Corrected author organization for William Denniss to Google

Changed definition of SET to be 2 parts, an envelope and 1 or more payloads.
Clarified that the intent is to express a single event with optional extensions only.

- mbj - Registered "events" claim, and proof-reading corrections.

Draft 04 - PH -

o Re-added the "sub" claim with clarifications that any SET type may use it.
o Added additional clarification on the use of envelope vs. payload attributes
o Added security consideration for event timing.
o Switched use of "attribute" to "claim" for consistency.
o Revised examples to put "sub" claim back in the top level.
o Added clarification that SETs typically do not use "exp".
o Added security consideration for distinguishing Access Tokens and SETs.

Draft 05 - PH - Fixed find/replace error that resulted in claim being spelled claimc

Draft 06 - PH -

o Corrected typos
o New txn claim
o New security considerations Sequencing and Timing Issues

Draft 07 -

o PH - Moved payload objects to be values of event URI attributes, per discussion.
o mbj - Applied terminology consistency and grammar cleanups.

Authors' Addresses

Phil Hunt (editor)
Oracle Corporation

Email: phil.hunt@yahoo.com

William Denniss
Google

Email: wdenniss@google.com

Morteza Ansari
Cisco

Email: morteza.ansari@cisco.com

Michael B. Jones
Microsoft

Email: mbj@microsoft.com
URI: http://self-issued.info/