Security Event Token (SET)
draft-ietf-secevent-token-01

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), which can be
   optionally signed and/or encrypted. A SET describes a statement of
   fact from the perspective of an issuer that it intends to share with
   one or more receivers.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on September 10, 2017.

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].

   This specification profiles the use of JWT for the purpose of issuing
   security event tokens (SETs). This specification defines a base
   format upon which profiling specifications define actual events and
   their meanings. Unless otherwise specified, this specification uses
   non-normative example events intended to demonstrate how events may
   be constructed.

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.

Security Events are not commands issued between parties. A security
event is a statement of fact from the perspective of an issuer about
the state of a security subject (e.g., a web resource, token, IP
address, the issuer itself) that the issuer controls or is aware of,
that has changed in some way (explicitly or implicitly). A security
subject MAY be permanent (e.g., a user account) or temporary (e.g.,
an HTTP session) in nature. A state change could describe a direct
change of entity state, an implicit change of state or other higher-
level security statements such as:

o The creation, modification, removal of a resource.

o The resetting or suspension of an account.

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

o The logout of a user session. Or,

o 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.

While subject state changes are often triggered by a user-agent or
security-subsystem, the issuance and transmission of an event often
occurs asynchronously and in a back-channel to the action which
caused the change that generated the security event. Subsequently,
an Event Receiver having received a SET, validates and interprets the
received SET and takes its own independent action, if any. For
example, having been informed of a personal identifier being
associated with a different security subject (e.g., an email address
is being used by someone else), the Event Receiver may choose to
ensure that the new user is not granted access to resources
associated with the previous user. Or, the Event Receiver may not
have any relationship with the subject, and no action is taken.

While Event Receivers will often take actions upon receiving SETs,
security events MUST NOT be assumed to be commands or requests. The
intent of this specification is to define a way of exchanging
statements of fact that subscribers may interpret for their own
purposes. As such, SETs have no capability for error signaling other
to ensure the validation of a received SET.

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:

   Security Event Token (SET)
      A SET is a JWT [RFC7519] that is distributed to one or more
      registered Event Receivers.

   Event Transmitter
      A service provider that delivers SETs to other providers known as
      Event Receivers.

   Event Receiver
      An Event Receiver is an entity that receives SETs through some
      distribution method.

   Subject
      A SET describes an event or state change that has occurred about a
      Subject. A Subject may be a principal (e.g., Section 4.1.2
      [RFC7519]), a web resource, an entity such as an IP address, or
      the issuer itself that a SET might reference.

   Profiling Specification A specification that uses the SET Token
      specification to define one or more event types and the associated
      claims included.

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 Subject.

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

   o An outer JSON object 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 Events within a single Profiling Specification or a
      related series of specifications. Claims in the envelope (the
      outer JSON structure) 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 a SET and declare the contents
      of the event data included in the SET. The claim "events"
      identifies the event types expressed that are related to the
      Security Subject and MAY also include event-specific data.

   o Each JSON member of the "events" object is a name and value pair.
      The JSON attribute name is a URI String value that expresses an
      event type, and the corresponding value is a JSON object known as
      the event "payload". The payload JSON object contains claims
      typically unique to the event's URI type value and are not
      registered as JWT claims. These claims are defined by their
      associated Profiling Specification. An event with no payload
      claims SHALL be represented as the empty JSON object ("{}"). In
      many cases, one event URI expresses the primary event URI, while
      other events might be considered extensions that MAY be used to do
      things such as:

      * A categorization event type to provide classification
         information (e.g., threat type or level).

      * An enhancement of an existing specifications the arise over
         time.

      * An extension needed to link a potential series of events.

      * Localized specific purpose event URI used between a particular
         Event Transmitter and Receiver.

The following is a non-normative example showing the JWT Claims Set
for a hypothetical SCIM password reset SET. This example shows an
additional events value ("https://example.com/scim/event/
passwordResetExt") 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:

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

o An "iss" claim, denoting the Event Transmitter.

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

o The "aud" claim specifies the intended audiences for the event.
   The syntax of the "aud" claim is defined in Section 4.1.3
   [RFC7519].

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 its own event URI attribute.

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/specs/consent.html":{
      "iss":"https://connect.example.com",
      "consentUri":[
        "https://terms.examplemed.com/labdisclosure.html#Agree"
      ]
    }
  }
}
```

                   Figure 3: Example Consent Event

In the above example, the attribute "iss" contained within the
payload for the event "https://openid.net/heart/specs/consent.html",
refers to the issuer of the Security Subject ("sub") rather than the
event issuer "https://my.examplemed.com". 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. Note
   that when a SET is expressing an event about a Security Subject
   for which the SET issuer is not the issuer of the Security
   Subject, the conflict SHALL be resolved by including the Security
   Subject "iss" value within the event "payload" (see "events"
   claim).

aud
   The syntax of the claim is as defined in Section 4.1.3 [RFC7519].
   This claim MAY contains one or more audience identifiers for the
   SET. 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 as equivalent
   to the actual time of the event. This claim is REQUIRED.

nbf
   Defined by Section 4.1.5 [RFC7519], a number whose value is a
   NumericDate. In the context of the SET token it SHALL be
   interpreted to mean a date in which the event is believed to have
   occurred (in the past) or will occur in the future. Note: there
   MAY be some cases where "nbf" is still smaller than "iat" such as
   when it took an extended time for a SET to be issued (for example
   after some analysis). 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
   The semantics of this claim is to define a set of event statements
   that each MAY add additional claims to fully describe a single
   logical event that has occurred (e.g. a state change to a
   subject). Multiple event statements of the same type SHALL NOT be
   accepted. The "events" claim SHOULD NOT be used to express
   multiple logical events.

   The value of "events" is a JSON object whose members are a set of
   JSON name/value pairs whose names are URIs representing the event
   statements being expressed. Event URI values SHOULD be stable
   values (e.g. a permanent URL for an event specification). 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
   Specification.

txn
   An OPTIONAL 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 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"]
    }
  }
}
```

                    Figure 4: Example Event Claims

When transmitted, the above JSON body must be converted into a JWT as
per [RFC7519].

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:

e3sgIAogICJqdGkiOiAiNGQzNTU5ZWM2NzUwNGFhYmE2NWQ0MGIwMzYzZmFhZDgiLAog
ICJpYXQiOiAxNDU4NDk2NDA0LAogICJpc3MiOiAiaHR0cHM6Ly9zY2ltLmV4YW1wbGUu
Y29tIiwgIAogICJhdWQiOiBbCiAgICJodHRwczovL3NjaW0uZXhhbXBsZS5jb20vRmVl
ZHMvOThkNTI0NjFmYTViYmM4Nzk1OTRiNzc1NCIsCiAgICJodHRwczovL3NjaW0uZXhh
bXBsZS5jb20vRmVlZHMvNWQ3NjA0NTE2YjFkMDg2NDFkNzY3NmVlNyIKICBdLAogICiAg
CiAgImV2ZW50cyI6IHsKICAgICJ1cm46aWV0ZjpwYXJhbXM6c2NpbTpldmVudDpjcmVh
dGUiOiB7CiAgICAgICJyZWYiOogICAgICAgICJodHRwczovL3NjaW0uZXhhbXBsZS5j
b20vVXNlcnMvNDRmNjE0MmRmOTZiZDZhYjYxZTc1MjFkOSIsCiAgICAgICJhdHRyaWJ1
dGVzIjpbImlkIiwgIm5hbWUiLCAidXNlck5hbWUiLCAicGFzc3dvcmQiLCAiZW1haWxz
Il0KICAgIH0KICB9Cn0

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

eyJhbGciOiJub25lIn0
.
e3sgIAogICJqdGkiOiAiNGQzNTU5ZWM2NzUwNGFhYmE2NWQ0MGIwMzYzZmFhZDgiLAog
ICJpYXQiOiAxNDU4NDk2NDA0LAogICJpc3MiOiAiaHR0cHM6Ly9zY2ltLmV4YW1wbGUu
Y29tIiwgIAogICJhdWQiOiBbCiAgICJodHRwczovL3NjaW0uZXhhbXBsZS5jb20vRmVl
ZHMvOThkNTI0NjFmYTViYmM4Nzk1OTRiNzc1NCIsCiAgICJodHRwczovL3NjaW0uZXhh
bXBsZS5jb20vRmVlZHMvNWQ3NjA0NTE2YjFkMDg2NDFkNzY3NmVlNyIKICBdLAogICiAg
CiAgImV2ZW50cyI6IHsKICAgICJ1cm46aWV0ZjpwYXJhbXM6c2NpbTpldmVudDpjcmVh
dGUiOiB7CiAgICAgICJyZWYiOogICAgICAgICJodHRwczovL3NjaW0uZXhhbXBsZS5j
b20vVXNlcnMvNDRmNjE0MmRmOTZiZDZhYjYxZTc1MjFkOSIsCiAgICAgICJhdHRyaWJ1
dGVzIjpbImlkIiwgIm5hbWUiLCAidXNlck5hbWUiLCAicGFzc3dvcmQiLCAiZW1haWxz
Il0KICAgIH0KICB9Cn0
.

Figure 5: Example Unsecured Security Event Token

For the purpose of a simpler example in Figure 5, an unencrypted
token was shown. When SETs are not signed or encrypted, the Event
Receiver MUST depend upon TLS and HTTP to authenticate the sender and
the security of the channel to authenticate the SET and its sender.

When validation (i.e. auditing), or additional transmission security
is required, JWS Signing and JWS Encryption MAY be used. 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
   Event Receiver.

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 encrypted SETs, without TLS there is no assurance that the
   correct endpoint received the SET and that it could be successfully
   processed.

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.

   Profiling 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 issuer 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:

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

o 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.

o 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.

o 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 Transmitters 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, Event
   Transmitters and Receivers MUST have the appropriate legal agreements
   and user consent or terms of service in place.

   The propagation of subject identifiers can be perceived as personally
   identifiable information. Where possible, Event Transmitters and
   Receivers 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

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

   o Claim Name: "txn"
   o Claim Description: Transaction Identifier
   o Change Controller: IESG
   o 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

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

   [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>.

   [saml-core-2.0]
              Internet2, "Assertions and Protocols for the OASIS
              Security Assertion Markup Language (SAML) V2.0", March
              2005.

Appendix A. Acknowledgments

   The editors would like to thank the members of the SCIM WG which
   began discussions of provisioning events starting with: draft-hunt-
   scim-notify-00 in 2015.

   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

   From the original draft-hunt-idevent-token:

   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.

Draft 08 - PH -

o Added clarification to status of examples

    o Changed from primary vs. extension to state that multiple events
       may be expressed, some of which may or may not be considered
       extensions of others (which is for the subscriber or profiling
       specifications to determine).
    o Other editorial changes suggested by Yaron
    From draft-ietf-secevent-token:

    Draft 00 - PH - First WG Draft based on draft-hunt-idevent-token

    Draft 01 - PH - Changes as follows:

    o Changed terminology away from pub-sub to transmitter/receiver
       based on WG feedback
    o Cleaned up/removed some text about extensions (now only used as
       example)
    o Clarify purpose of spec vs. future profiling specs that define
       actual events

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/