secevent
Internet-Draft
Intended status: Informational
Expires: May 25, 2018

A. Backman
Amazon
W. Denniss
Google
M. Ansari
Cisco
M. Jones
Microsoft

November 21, 2017

Security Event Token (SET)
draft-backman-secevent-token-00

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 https://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 May 25, 2018.

Table of Contents

1. Introduction

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:

- 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.

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 actions, 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 cannot be assumed to be commands or requests. The intent of this specification is to define a way of exchanging statements of fact that Event Receivers 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", "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 contains an event payload describing a security event.

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. An Event Receiver is the same entity referred as "recipient" or "receiver" in and related specifications. [RFC7519]

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 is a data structure that encodes an "event payload" describing a security event, wrapped in an "envelope" providing metadata and context for the security event. The SET envelope is a JWT Claims Set as defined in [RFC7519], consisting of a JSON object containing a set of claims. The event payload is a JSON object contained within the

SET envelope, itself containing claims that express information about the event, e.g. the type of event, the subject of the event, and other information defined in a Profiling Specification.

This specification defines a core set of claims for use in SET envelopes and event payloads, however Profiling Specifications MAY define additional claims of both types. It is RECOMMENDED that Profiling Specifications define claims to be used in the event payload rather than the envelope. If a Profiling Specification does define envelope claims, those claims SHOULD be registered in the JWT Token Claims Registry [IANA.JWT.Claims] or have Public Claim Names as defined in Section 4.2 of [RFC7519].

2.1. SET Claims

This specification profiles the following claims defined in [RFC7519] for use in the SET envelope:

iss
: A case-sensitive string identifying the principal that issued the SET, as defined by Section 4.1.1 of [RFC7519]. This claim is REQUIRED.

aud

: A case-sensitive string or array of case-sensitive strings identifying the audience for the SET, as defined by Section 4.1.3 of [RFC7519]. This claim is RECOMMENDED.

exp
: As defined by Section 4.1.4 of [RFC7519], this claim is the time after 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. Use of this claim is NOT RECOMMENDED.

iat
: A value identifying the time at which the SET was issued, as defined by Section 4.1.6 of [RFC7519]. This claim is REQUIRED.

jti
: A unique identifier for an event, as defined by Section 4.1.7 of [RFC7519]. This claim is REQUIRED.

This specification defines the following new claims for use in the SET envelope:

event
: A JSON object known as the "event payload", whose contents identify the type of event contained within the SET and contain additional information defined as part of an event type definition in a Profiling Specification.

This specification defines the following claims for use in event payloads:

event_type
: A string containing a URI that uniquely identifies an event type defined by a Profiling Specification. This claim is REQUIRED.

event_id
: A string that identifies a specific "real world" event or state change to which this event is related. Recipients MAY use this claim to correlate events across different SETs received at different times and/or by different systems. The value of this claim MUST be unique with respect to the transmitter to a specific "real world" event or state change, however recipients MUST NOT interpret a difference in "event_id" values as a guarantee that two events are not related. This claim is OPTIONAL.

event_subject
: A Subject Identifier that identifies the subject of the event. (See: Section 2.2) This claim is REQUIRED.

event_time
: A number identifying the date and time at which the event is believed to have occurred or will occur in the future. Its value MUST take the form of a NumericDate value, as defined in Section 2 of [RFC7519]. This claim is OPTIONAL, however if it is not present then the recipient MUST interpret that to mean that no event time is being asserted, either because there is no specific event time, the transmitter does not wish to share it, or the

transmitter does not know its value.

Both the SET envelope and event payload MAY contain additional claims, such as those defined in a Profiling Specification. The format and meaning of these claims is out of scope of this specification. Implementations SHOULD ignore any claims in the SET envelope or event payload that they do not understand.

The following is a non-normative example showing a SET envelope expressing a hypothetical event with two additional claims in the event payload:

```
{
  "jti": "3d0c3cf797584bd193bd0fb1bd4e7d30",
  "iss": "https://transmitter.example.com",
  "aud": [ "https://receiver.example.com" ],
  "iat": 1458496025,
  "event": {
    "event_type": "https://secevent.example.com/example_event",
    "event_subject": {
      "identifier_type": "urn:ietf:params:secevent:subject:email",
      "email": "user@example.com"
    },
    "event_time": 1458492425,
    "claim_1": "foo",
    "claim_2": "bar"
  }
}
```

Figure 1: Example SET With Event Claims In Payload

The payload in this example contains the following:

- An "event_type" claim whose value is the URI identifying the hypothetical event type.
- An "event_subject" claim whose value identifies a subject via email address.

- An "event_time" claim whose value is the time at which the event

occured.

- Two claims "claim_1" and "claim_2" that are defined by the hypothetical event type's Profiling Specification.

2.2. Subject Identifiers

The Subject Identifier provides a common syntax for expressing the subject of a security event. A Subject Identifier is a JSON object representing an instance of a Subject Identifier Type. A Subject Identifier Type defines a way of identifying the subject of an event. Typically this is done by defining a set of one or more claims about a subject that when taken together collectively identify that subject. Each Subject Identifier Type MUST have a name which MUST be registered in the IANA "SET Subject Identifier Types" registry established by Section 7.1.

A Subject Identifier MUST contain an "identifier_type" claim, whose value is a string containing the name of the Subject Identifier's Subject Identifier Type. All other claims within the Subject Identifier MUST be defined by the Subject Identifier Type.

The names of the Subject Identifier Types defined below are registered in the IANA "SET Subject Identifier Types" registry established by Section 7.1.

2.2.1. Implicit Subject Identifier Type

The "Implicit" Subject Identifier Type indicates that the recipient is to be determined implicitly, either from other claims in the SET envelope or event payload, or through some other context. For example, there may be event types for which the only logical subject is the transmitter itself, in which case the subject is implicitly known from the "iss" claim in the SET envelope.

The Implicit Subject Identifier Type has the name "implicit". This type contains no additional claims.

The following is a non-normative example of a Subject Identifier representing an instance of the Implicit Subject Identifier Type:

```
{
  "identifier_type": "implicit"
}
```

Figure 2: An Instance of the Implicit Subject Identifier Type

2.2.2. Email Subject Identifier Type

The "Email" Subject Identifier Type identifies a subject by email address. It has the name "email", and contains a single additional claim:

email
: A string containing an email address. Its value SHOULD conform to [RFC5322]. This claim is REQUIRED.

The following is a non-normative example of a Subject Identifier representing an instance of the Email Subject Identifier Type:

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

Figure 3: An Instance of the Email Subject Identifier Type

2.2.3. Phone Number Subject Identifier Type

The "Phone Number" Subject Identifier Type identifies a subject by phone number. It has the name "phone_number", and contains a single claim:

phone_number
: A string containing a phone number. It SHOULD be formatted according to [E.164]. This claim is REQUIRED.

The following is a non-normative example of a Subject Identifier representing an instance of the Phone Number Subject Identifier Type:

```
{
  "identifier_type": "phone_number",
  "phone_number": "+1 206 555 0123"
}
```

Figure 4: An Instance of the Phone Number Subject Identifier Type

2.2.4. Issuer and Subject Subject Identifier Type

The "Issuer and Subject" Subject Identifier Type identifies a subject by an issuer and subject pair. It has the name "iss-sub", and contains two claims:

iss

A case-sensitive string identifying the principal who is responsible for assignment of the identifier in the "sub" claim, as defined by Section 4.1.1 of [RFC7519]. This claim is REQUIRED.

sub
A case-sensitive string containing an identifier that identifies a subject within the context of the principal identified by the "iss" claim, as defined by Section 4.1.2 of [RFC7519]. This claim is REQUIRED.

The following is a non-normative example of a Subject Identifier representing an instance of the Issuer and Subject Subject Identifier Type:

```
{
  "identifier_type": "iss-sub",
  "iss": "http://id.example.com",
  "sub": "example.user.1234"
}
```

Figure 5: An Instance of the Issuer and Subject Subject Identifier Type

2.3. Explicit Typing of SETs

This specification registers the "application/secevent+jwt" media type. SETs MAY include this media type in the "typ" header parameter of the JWT containing the SET to explicitly declare that the JWT is a SET. This MUST be included if the SET could be used in an application context in which it could be confused with other kinds of JWTs. Profiling Specifications MAY declare that this is REQUIRED for SETs containing events defined by the Profiling Specification.

Per the definition of "typ" in Section 4.1.9 of [RFC7515], it is RECOMMENDED that the "application/" prefix be omitted. Therefore, the "typ" value used SHOULD be "secevent+jwt".

2.4. Security Event Token Construction

A SET is a JWT, and therefore it's construction follows that described in [RFC7519].

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

The following is the example JWT Claims Set from Figure 1, expressed as an unsigned JWT. The JOSE Header is:

{"typ":"secevent+jwt","alg":"none"}

Base64url encoding of the octets of the UTF-8 representation of the JOSE Header yields:

eyJ0eXAiOiJzZWNldmVudCtqd3QiLCJhbGciOiJub25lIn0

The example JWT Claims Set is encoded as follows:

ew0KICAgImp0aSI6ICIzZDBjM2NmNzk3NTg0YmQxOTNiZDBmYjFiZDRlN2QzMCIsDQog
ICAiaXNzIjogImh0dHBzOi8vdHJhbnNtaXR0ZXIuZXhhbXBsZS5jb20iLA0KICAgImF1
ZCI6IFsgImh0dHBzOi8vcmVjZWl2ZXIuZXhhbXBsZS5jb20iIF0sDQogICAiaWF0Ijog
MTQ1ODQ5NjAyNSwNCiAgICJldmVudCI6IHsNCiAgICAgImV2ZW50X3R5cGUiOiAiaHR0
cHM6Ly9zZWNldmVudC5leGFtcGxlLmNvbS9leGFtcGxlX2V2ZW50IiwNCiAgICAgImV2
ZW50X3N1YmplY3QiOiB7DQogICAgICAgImlkZW50aWZpZXJfdHlwZSI6ICJlbWFpbCIs
DQogICAgICAgImVtYWlsIjogInVzZXJAZXhhbXBsZS5jb20iDQogICAgIH0sDQogICAg
ICJldmVudF90aW1lIjogMTQ1ODQ5MjQyNSwNCiAgICAgImNsYWltXzEiOiAiZm9vIiwN
CiAgICAgImNsYWltXzIiOiAiYmFyIg0KICAgfQ0KIH0=

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

eyJ0eXAiOiJzZWNldmVudCtqd3QiLCJhbGciOiJub25lIn0.
ew0KICAgImp0aSI6ICIzZDBjM2NmNzk3NTg0YmQxOTNiZDBmYjFiZDRlN2QzMCIsDQog
ICAiaXNzIjogImh0dHBzOi8vdHJhbnNtaXR0ZXIuZXhhbXBsZS5jb20iLA0KICAgImF1
ZCI6IFsgImh0dHBzOi8vcmVjZWl2ZXIuZXhhbXBsZS5jb20iIF0sDQogICAiaWF0Ijog
MTQ1ODQ5NjAyNSwNCiAgICJldmVudCI6IHsNCiAgICAgImV2ZW50X3R5cGUiOiAiaHR0
cHM6Ly9zZWNldmVudC5leGFtcGxlLmNvbS9leGFtcGxlX2V2ZW50IiwNCiAgICAgImV2
ZW50X3N1YmplY3QiOiB7DQogICAgICAgImlkZW50aWZpZXJfdHlwZSI6ICJlbWFpbCIs
DQogICAgICAgImVtYWlsIjogInVzZXJAZXhhbXBsZS5jb20iDQogICAgIH0sDQogICAg
ICJldmVudF90aW1lIjogMTQ1ODQ5MjQyNSwNCiAgICAgImNsYWltXzEiOiAiZm9vIiwN
CiAgICAgImNsYWltXzIiOiAiYmFyIg0KICAgfQ0KIH0=.

Figure 6: Example Unsecured Security Event Token

For the purpose of a simpler example in Figure 5, an unsecured token was shown. When SETs are not signed or encrypted, the Event Receiver MUST employ other mechanisms such as TLS and HTTP to provide integrity, confidentiality, and issuer validation, as needed by the application.

When validation (i.e. auditing), or additional transmission security is required, JWS signing and/or JWE encryption MAY be used. To create and or validate a signed and/or encrypted SET, follow the instructions in Section 7 of [RFC7519].

3. Related Events

In order to accommodate use cases that require communicating multiple related security events to an Event Receiver, this section defines the "Related Events" event type. A Related Events event is essentially a container for two or more events that are related to one another, in that they represent or express different aspects of the same event or state change. The Related Events event SHOULD NOT be used to combine unrelated events into a single set, and MUST NOT be used as a general purpose batch transmission mechanism. Profiling Specifications that require an event grouping mechanism with these or other semantics are encouraged to define additional event types for their use cases.

The event type for the Related Events event is the URN "urn:ietf:secevents:related_events".

The Related Events event has a single additional event payload claim:

events
: An array of event payloads, as defined in this document. These event payloads can be referred to as Nested Events for the Related Events event. This claim is REQUIRED.

3.1. Processing Related Events

Nested Events can inherit the "event_id", "event_subject", and "event_time" claims from the Related Events payload. Transmitters MAY omit some, all, or none of these claims from a Nested Event. Transmitters MAY omit claims from some Nested Events and include them in others within the same Related Events event. When a claim is omitted, recipients MUST use the value of the corresponding claim in the Related Event event's payload.

The following is a non-normative example of a SET containing a Related Events event:


```
{
  "jti": "1c0038c2-02db-40de-ad50-122a64724166",
  "iss": "https://transmitter.example.com",
  "aud": [ "https://receiver.example.com" ],
  "iat": 1510666261,
  "event": {
    "event_type": "urn:ietf:secevent:related_events",
    "event_subject": {
      "identifier_type": "email",
      "email": "user@example.com"
    },
    "event_id": "container",
    "event_time": 1510662661,
    "events": [
      {
        "event_id": "nested_1",
        "event_type": "http://specs.example.com/set_profile/event_1"
      },
```

```
      {
        "event_id": "nested_2",
        "event_type": "http://specs.example.com/set_profile/event_2",
        "event_time": 151059061
      }
    ]
  }
}
```

Figure 7: Example SET Containing A Related Events Event

The following table demonstrates how Nested Events inherit values for omitted claims:

Event ID	Event Time	Event Subject
container	151062661	{ "identifier_type": "email", "email": "user@example.com" }
nested_1		
nested_2	151059061	

Figure 8: Example of Event Payloads Inheriting Values for Omitted Claims

Since the Nested Event with event ID "nested_1" omits the "event_time" claim, it inherits the event time from the Related Events event payload. Similarly, since both Nested Events "nested_1" and "nested_2" omit the "event_subject" claim, both inherit the event subject from the Related Events event payload.

4. Requirements for SET Profiles

Profiling Specifications for SETs define the syntax and semantics of SETs conforming to that SET profile and rules for validating those SETs. The syntax defined by Profiling Specifications includes what SET envelope and event payload claims are used by SETs expressing and event defined by the profile.

Defining the semantics of the SET contents for SETs utilizing the profile is equally important. Possibly most important is defining the procedures used to validate the SET issuer and to obtain the keys controlled by the issuer that were used for cryptographic operations used in the JWT representing the SET. For instance, some profiles may define an algorithm for retrieving the SET issuer's keys that uses the "iss" claim value as its input. Likewise, if the profile allows (or requires) that the JWT be unsecured, the means by which the integrity of the JWT is ensured MUST be specified.

Profiling Specifications MUST define how the event Subject is identified in the SET, as well as how to differentiate between the event Subject's Issuer and the SET Issuer, if applicable. It is NOT RECOMMENDED for Profiling Specifications to use the "sub" claim defined in [RFC7519] in cases in which the Subject is not globally unique and has a different Issuer from the SET itself.

Profiling Specifications MUST clearly specify the steps that a recipient of a SET utilizing that profile MUST perform to validate that the SET is both syntactically and semantically valid.

4.1. Extending Events

As needs change and new use cases develop, it may be desirable to augment existing event definitions with new claims. In order to avoid collisions, Profiling Specifications that extend existing events with additional event payload claims SHOULD use Collision-Resistant Names as defined in Section 2 of [RFC7519] for the names of the new claims.

5. Security Considerations

5.1. 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.

Unless integrity of the JWT is ensured by other means, it MUST be signed using JWS [RFC7515] so that individual events can be authenticated and validated by the Event Receiver.

5.2. Delivery

This specification does not define a delivery mechanism by itself. In addition to confidentiality and integrity (discussed above), implementers and Profiling 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.

5.3. 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.

5.4. 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 an Event Receiver 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.

5.5. Distinguishing SETs from ID 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 with ID Token [OpenID.Core] if a SET is mistakenly or intentionally used in a context requiring an ID Token. If a SET could otherwise be interpreted as a valid ID Token (because it includes the required claims for an ID Token and valid issuer and audience claim values for an ID Token) then that SET profile MUST require that the "exp" claim not be present in the SET. Because "exp" is a required claim in ID Tokens, valid ID Token implementations will reject such a SET if presented as if it were an ID Token.

Excluding "exp" from SETs that could otherwise be confused with ID Tokens is actually defense in depth. In any OpenID Connect contexts in which an attacker could attempt to substitute a SET for an ID Token, the SET would actually already be rejected as an ID Token because it would not contain the correct "nonce" claim value for the ID Token to be accepted in contexts for which substitution is possible.

Note that the use of explicit typing, as described in Section 2.2, will not achieve disambiguation between ID Tokens and SETs, as the ID Token validation rules do not use the "typ" header parameter value.

5.6. Distinguishing SETs from Access Tokens

OAuth 2.0 [RFC6749] defines access tokens as being opaque. Nonetheless, some implementations implement access tokens as JWTs.

Because the structure of these JWTs is implementation-specific,

ensuring that a SET cannot be confused with such an access token is therefore likewise, in general, implementation specific. Nonetheless, it is recommended that SET profiles employ the following strategies to prevent possible substitutions of SETs for access tokens in contexts in which that might be possible:

- Prohibit use of the "exp" claim, as is done to prevent ID Token confusion.

- Where possible, use a separate "aud" claim value to distinguish between the Event Receiver and the protected resource that is the audience of an access token.

- Modify access token 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.

- Employ explicit typing, as described in Section 2.2, and modify access token validation systems to use the "typ" header parameter value.

5.7. Distinguishing SETs from other kinds of JWTs

JWTs are now being used in application areas beyond the identity applications in which they first appeared. For instance, the Session Initiation Protocol (SIP) Via Header Field [RFC8055] and Personal Assertion Token (PASSporT) [I-D.ietf-stir-passport] specifications both define JWT profiles that use mostly or completely different sets of claims than are used by ID Tokens. If it would otherwise be possible for an attacker to substitute a SET for one of these (or other) kinds of JWTs, then the SET profile must be defined in such a way that any substituted SET will result in its rejection when validated as the intended kind of JWT.

The most direct way to prevent confusion is to employ explicit typing, as described in Section 2.2, and modify applicable token validation systems to use the "typ" header parameter value. This approach can be employed for new systems but may not be applicable to existing systems.

Another way to ensure that a SET is not confused with another kind of JWT is to have the JWT validation logic reject JWTs containing an "events" claim unless the JWT is intended to be a SET. This approach can be employed for new systems but may not be applicable to existing systems.

For many use cases, the simplest way to prevent substitution is requiring that the SET not include claims that are required for the kind of JWT that might be the target of an attack. For example, for [RFC8055], the "sip_callid" claim could be omitted and for [I-D.ietf-stir-passport], the "orig" claim could be omitted.

In many contexts, simple measures such as these will accomplish the task, should confusion otherwise even be possible. Note that this topic is being explored in a more general fashion in JSON Web Token Best Current Practices [I-D.sheffer-oauth-jwt-bcp]. The proposed best practices in that draft may also be applicable for particular SET profiles and use cases.

6. 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 an Event Receiver.

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 Event Receiver to know the subject.

7. IANA Considerations

7.1. SET Subject Identifier Types Registry

This section establishes the IANA "SET Subject Identifier Types" registry // TODO

7.2. JSON Web Token Claims Registration

This specification registers the "event" claim in the IANA "JSON Web Token Claims" registry [IANA.JWT.Claims] established by [RFC7519].

7.2.1. Registry Contents

- Claim Name: "event"
- Claim Description: Security Event Payload
- Change Controller: IESG
- Specification Document(s): Section 2.1 of [[this specification]]

7.3. Media Type Registration

7.3.1. Registry Contents

This section registers the "application/secevent+jwt" media type [RFC2046] in the "Media Types" registry [IANA.MediaTypes] in the manner described in [RFC6838], which can be used to indicate that the content is a SET.

- Type name: application
- Subtype name: secevent+jwt
- Required parameters: n/a
- Optional parameters: n/a

- Encoding considerations: 8bit; A SET is a JWT; JWT values are encoded as a series of base64url-encoded values (some of which may be the empty string) separated by period ('.') characters.

- Security considerations: See the Security Considerations section of [[this specification]]

- Interoperability considerations: n/a

- Published specification: Section 2.2 of [[this specification]]

- Applications that use this media type: TBD

- Fragment identifier considerations: n/a

- Additional information:

- Magic number(s): n/a

- File extension(s): n/a

- Macintosh file type code(s): n/a

- Person & email address to contact for further information: Michael B. Jones, mbj@microsoft.com

- Intended usage: COMMON

- Restrictions on usage: none

- Author: Michael B. Jones, mbj@microsoft.com

- Change controller: IESG

- Provisional registration? No

8. References

8.1. Normative References

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

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

[IANA.MediaTypes] IANA, "Media Types", n.d., <http://www.iana.org/assignments/media-types>.

[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>.

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

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

[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, <https://www.rfc-editor.org/info/rfc6125>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework",

RFC 6749, DOI 10.17487/RFC6749, October 2012, <https://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, <https://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, <https://www.rfc-editor.org/info/rfc7525>.

8.2. Informative References

[I-D.ietf-stir-passport]
Wendt, C. and J. Peterson, "Personal Assertion Token (PASSporT)", draft-ietf-stir-passport-11 (work in progress), February 2017.

[I-D.sheffer-oauth-jwt-bcp]
Sheffer, Y., Hardt, D., and M. Jones, "JSON Web Token Best Current Practices", draft-sheffer-oauth-jwt-bcp-01 (work in progress), July 2017.

[OpenID.Core]
"OpenID Connect Core 1.0", n.d., <http://openid.net/specs/openid-connect-core-1_0.html>.

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, DOI 10.17487/RFC2046, November 1996, <https://www.rfc-editor.org/info/rfc2046>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>.

[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>.

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

[RFC8055] Holmberg, C. and Y. Jiang, "Session Initiation Protocol (SIP) Via Header Field Parameter to Indicate Received Realm", RFC 8055, DOI 10.17487/RFC8055, January 2017, <https://www.rfc-editor.org/info/rfc8055>.

Appendix A. Acknowledgments

The editors would like to thank the participants on the IETF secevent mailing list and related working groups for their support of this specification.

Authors' Addresses

Annabelle Backman
Amazon

Email: richanna@amazon.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/