Internet Engineering Task Force (IETF)
Request for Comments: 8417
Category: Standards Track
ISSN: 2070-1721

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

July 2018

Security Event Token (SET)

Abstract

This specification defines the Security Event Token (SET) data structure. A SET describes statements of fact from the perspective of an issuer about a subject. These statements of fact represent an event that occurred directly to or about a security subject, for example, a statement about the issuance or revocation of a token on behalf of a subject. This specification is intended to enable representing security- and identity-related events. A SET is a JSON Web Token (JWT), which can be optionally signed and/or encrypted. SETs can be distributed via protocols such as HTTP.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8417.

Table of Contents

1. Introduction and Overview

This specification defines an extensible Security Event Token (SET) data structure, which can 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 SETs. This specification defines a base format used by profiling specifications to define actual events and their meanings. This specification uses non-normative example events to demonstrate how events can be constructed.

This specification is scoped to security- and identity-related events. While SETs 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 SET describes statements of fact from the perspective of an issuer about a subject (e.g., a web resource, token, IP address, the issuer itself). These statements of fact represent a logical event that occurred directly to or about a security subject, for example, a statement about the issuance or revocation of a token on behalf of a subject. 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.
- An indication that a user has been given control of an email identifier that was previously controlled by another user.

While subject state changes are often triggered by a user agent or security subsystem, the issuance and transmission of an event may occur asynchronously and in a back channel to the action that caused the change that generated the security event. Subsequently, a SET recipient, 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 SET recipient may choose to ensure that the new user is not granted access to resources associated with the previous user. Or, the SET recipient may not have any relationship with the subject, and no action is taken.

While SET recipients 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 syntax for statements of fact that SET recipients may interpret for their own purposes.

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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] conforming to this specification.

SET Issuer
: A service provider that creates SETs to be sent to other service providers known as SET recipients.

SET Recipient
: A SET recipient is an entity that receives SETs through some distribution method. A SET recipient is the same entity referred as a "recipient" in [RFC7519] or "receiver" in related specifications.

Subject
: A SET describes an event or state change that has occurred to a subject. A subject might, for instance, be a principal (e.g., Section 4.1.2 of [RFC7519]), a web resource, an entity such as an IP address, or the issuer of the SET.

Event Identifier
: A member name for an element of the JSON object that is the value of the "events" claim in a SET. This member name MUST be a URI.

Event Payload
: A member value for an element of the JSON object that is the value of the "events" claim in a SET. This member value MUST be a JSON object.

Profiling Specification
: A specification that profiles the SET data structure to define one or more specific event types and their associated claims and processing rules.

2. The Security Event Token (SET)

A SET is a JWT [RFC7519] data structure that represents one or more related aspects of a security event that occurred to a subject. The JWT Claims Set in a SET has the following structure:

- The top-level claims in the JWT Claims Set are called the SET "envelope". Some of these claims are present in every SET; others will be specific to particular SET profiles or profile families. Claims in the envelope SHOULD be registered in the "JSON Web Token Claims" registry [IANA.JWT.Claims] or be Public Claims or Private Claims, as defined in [RFC7519].

- Envelope claims that are profiled and defined in this specification are used to validate the SET and provide information about the event data included in the SET. The "events" claim contains the event identifiers and event-specific data expressed about the security subject. The envelope MAY include event-specific or profile-specific data. The "events" claim value MUST be a JSON object that contains at least one member.

- Each member of the "events" JSON object is a name/value pair. The JSON member name is a URI string value, which is the event identifier, and the corresponding value is a JSON object known as the event "payload". The payload JSON object contains claims that pertain to that event identifier and need not be registered as JWT

claims. These claims are defined by the profiling specification that defines the event. An event with no payload claims SHALL be represented as the empty JSON object ("{}").

- When multiple event identifiers are contained in a SET, they represent multiple aspects of the same state transition that occurred to the security subject. They are not intended to be used to aggregate distinct events about the same subject. Beyond this, the interpretation of SETs containing multiple event identifiers is out of scope for this specification; profiling specifications MAY define their own rules regarding their use of SETs containing multiple event identifiers, as described in Section 3. Possible uses of multiple values include, but are not limited to:

  - Values to provide classification information (e.g., threat type or level).

  - Additions to existing event representations.

  - Values used to link potential series of events.

  - Specific-purpose event URIs used between particular SET issuers and SET recipients.

2.1. Illustrative Examples

This section illustrates several possible uses of SETs through non-normative examples.

2.1.1. SCIM Example

The following example shows the JWT Claims Set for a hypothetical System for Cross-domain Identity Management (SCIM) [RFC7644] password reset SET. Such a SET might be used by a receiver as a trigger to reset active user-agent sessions related to the identified user.

```
{
  "iss": "https://scim.example.com",
  "iat": 1458496025,
  "jti": "3d0c3cf797584bd193bd0fb1bd4e7d30",
  "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 JWT Claims Set usage consists of:

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

- The "iss" claim, denoting the SET issuer.

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

- The "aud" claim, specifying the intended audiences for the event. (The syntax of the "aud" claim is defined in Section 4.1.3 of [RFC7519].)

The SET contains two event payloads:

- The "id" claim represents SCIM’s unique identifier for a subject.

- The second payload identified by "https://example.com/scim/event/passwordResetExt" and the payload claim "resetAttempts" conveys the current count of reset attempts. In this example, while the count is a simple factual statement for the issuer, the meaning of the value (a count) is up to the receiver. As an example, such a value might be used by the receiver to infer increasing risk.

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 in the event payload of an event used to convey this information.

2.1.2. Logout Example

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 claims "sub" and "sid" to identify the subject that was logged out. At the time of this writing, this example corresponds to the logout token defined in the OpenID Connect Back-Channel Logout 1.0 [OpenID.BackChannel] specification.

2.1.3. Consent Example

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:

```
{
  "iss": "https://my.med.example.org",
  "iat": 1458496025,
  "jti": "fb4e75b5411e4e19b6c0fe87950f7749",
  "aud": [
    "https://rp.example.com"
  ],
  "events": {
    "https://openid.net/heart/specs/consent.html": {
      "iss": "https://connect.example.com",
      "sub": "248289761001",
      "consentUri": [
        "https://terms.med.example.org/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 SET issuer "https://my.med.example.org". 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.

2.1.4. RISC Example

The following example JWT Claims Set is for an account disabled event. At the time of this writing, this example corresponds to the account disabled event defined in the OpenID RISC Event Types 1.0 [OpenID.RISC.Events] specification.

```
{
  "iss": "https://idp.example.com/",
  "jti": "756E69717565206964656E746966696572",
  "iat": 1508184845,
  "aud": "636C69656E745F6964",
  "events": {
"https://schemas.openid.net/secevent/risc/event-type/account-disabled"
        : {
      "subject": {
        "subject_type": "iss-sub",
        "iss": "https://idp.example.com/",
        "sub": "7375626A656374"
      },
      "reason": "hijacking"
    }
  }
}
```

Figure 4: Example RISC Event

Notice that parameters to the event are included in the event payload, in this case, the "reason" and "cause-time" values. The subject of the event is identified using the "subject" payload value, which itself is a JSON object.

2.2. Core SET Claims

The following claims from [RFC7519] are profiled for use in SETs:

"iss" (Issuer) Claim
: As defined by Section 4.1.1 of [RFC7519], this claim contains a string identifying the service provider publishing the SET (the issuer). In some cases, the issuer of the SET will not be the issuer associated with the security subject of the SET. Therefore, implementers cannot assume that the issuers are the same unless the profiling specification specifies that they are for SETs conforming to that profile. This claim is REQUIRED.

"iat" (Issued At) Claim
As defined by Section 4.1.6 of [RFC7519], this claim contains a value representing when the SET was issued. This claim is REQUIRED.

"jti" (JWT ID) Claim
As defined by Section 4.1.7 of [RFC7519], this claim contains a unique identifier for the SET. The identifier MUST be unique within a particular event feed and MAY be used by clients to track whether a particular SET has already been received. This claim is REQUIRED.

"aud" (Audience) Claim
As defined by Section 4.1.3 of [RFC7519], this claim contains one or more audience identifiers for the SET. This claim is RECOMMENDED.

"sub" (Subject) Claim
As defined by Section 4.1.2 of [RFC7519], this claim contains a StringOrURI value representing the principal that is the subject of the SET. This is usually the entity whose "state" was changed. For example:

* an IP Address was added to a blacklist;

* a URI representing a user resource that was modified; or,

* a token identifier (e.g. "jti") for a revoked token.

If used, the profiling specification MUST 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. Note that some SET profiles MAY choose to convey event subject information in the event payload (either using the "sub" member name or another name), particularly if the subject information is relative to issuer information that is also conveyed in the event payload, which may be the case for some identity SET profiles.

"exp" (Expiration Time) Claim
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 typically apply, since a SET represents something that has already occurred and is historical in nature. Therefore, its use is NOT RECOMMENDED. (Also, see Section 4.1 for additional reasons not to use the "exp" claim in some SET use cases.)

The following new claims are defined by this specification:

"events" (Security Events) Claim
: This claim contains a set of event statements that each provide information describing a single logical event that has occurred about a security subject (e.g., a state change to the subject). Multiple event identifiers with the same value MUST NOT be used. The "events" claim MUST NOT be used to express multiple independent logical events.

  The value of the "events" claim is a JSON object whose members are name/value pairs whose names are URIs identifying the event statements being expressed. Event identifiers SHOULD be stable values (e.g., a permanent URL for an event specification). For each name present, the corresponding value MUST be a JSON object. The JSON object MAY be an empty object ("{}"), or it MAY be a JSON object containing data described by the profiling specification.

"txn" (Transaction Identifier) Claim
: An OPTIONAL string value that represents a unique transaction identifier. In cases in which multiple related JWTs are issued, the transaction identifier claim can be used to correlate these related JWTs. Note that this claim can be used in JWTs that are SETs and also in JWTs using non-SET profiles.

"toe" (Time of Event) Claim
: A value that represents the date and time at which the event occurred. This value is a NumericDate (see Section 2 of [RFC7519]). By omitting this claim, the issuer indicates that they are not sharing an event time with the recipient. (Note that in some use cases, the represented time might be approximate; statements about the accuracy of this field MAY be made by profiling specifications.) This claim is OPTIONAL.

2.3. Explicit Typing of SETs

This specification registers the "application/secevent+jwt" media type, which can be used to indicate that the content is a SET. SETs MAY include this media type in the "typ" header parameter of the JWT representing 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.

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

This section describes how to construct a SET.

The following is an example JWT Claims Set for a hypothetical SCIM SET:

```
{
  "iss": "https://scim.example.com",
  "iat": 1458496404,
  "jti": "4d3559ec67504aaba65d40b0363faad8",
  "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 5: Example Event Claims

The JSON Claims Set is encoded per [RFC7519].

In this example, the SCIM SET claims are encoded in an unsecured JWT. The JOSE Header for this example is:

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

Base64url encoding (as defined by Section 2 of [RFC7515], including the omission of all trailing '=' characters) of the octets of the UTF-8 [RFC3629] representation of the JOSE Header yields:

```
eyJ0eXAiOiJzZWNldmVudCtqd3QiLCJhbGciOiJub25lIn0
```

The above example JWT Claims Set (with insignificant whitespace removed) is encoded as follows (with line breaks for display purposes only):

```
eyJpc3MiOiJodHRwczovL3NjaW0uZXhhbXBsZS5jb20iLCJpYXQiOjE0NTg0OTY0M
DQsImp0aSI6IjRkMzU1OWVjNjc1MDRhYWJhNjVkNDBiMDM2M2ZhYWQ4IiwiYXVkIj
pbImh0dHBzOi8vc2NpbS5leGFtcGxlLmNvbS9GZWVkcy85OGQ1MjQ2MWZhNWJiYzg
3OTU5M2I3NzU0IiwiaHR0cHM6Ly9zY2ltLmV4YW1wbGUuY29tL0ZlZWRzLzVkNzYw
NDUxNmIxZDA4NjQxZDc2NzZlZTciXSwiZXZlbnRzIjp7InVybjppZXRmOnBhcmFtc
zpzY2ltOmV2ZW50OmNyZWF0ZSI6eyJyZWYiOiJodHRwczovL3NjaW0uZXhhbXBsZS
5jb20vVXNlcnMvNDRmNjE0MmRmOTZiZDZhYjYxZTc1MjFkOSIsImF0dHJpYnV0ZXM
iOlsiaWQiLCJuYW1lIiwidXNlck5hbWUiLCJwYXNzd29yZCIsImVtYWlscyJdfX19
```

The encoded JWS signature is the empty string.

Concatenating the three encoded parts (JOSE Header, JWT Claims Set, and JWS signature) in order with period ('.') characters between the parts yields this complete SET (with line breaks for display purposes only):

```
eyJ0eXAiOiJzZWNldmVudCtqd3QiLCJhbGciOiJub25lIn0
.
eyJpc3MiOiJodHRwczovL3NjaW0uZXhhbXBsZS5jb20iLCJpYXQiOjE0NTg0OTY0M
DQsImp0aSI6IjRkMzU1OWVjNjc1MDRhYWJhNjVkNDBiMDM2M2ZhYWQ4IiwiYXVkIj
pbImh0dHBzOi8vc2NpbS5leGFtcGxlLmNvbS9GZWVkcy85OGQ1MjQ2MWZhNWJiYzg
3OTU5M2I3NzU0IiwiaHR0cHM6Ly9zY2ltLmV4YW1wbGUuY29tL0ZlZWRzLzVkNzYw
NDUxNmIxZDA4NjQxZDc2NzZlZTciXSwiZXZlbnRzIjp7InVybjppZXRmOnBhcmFtc
zpzY2ltOmV2ZW50OmNyZWF0ZSI6eyJyZWYiOiJodHRwczovL3NjaW0uZXhhbXBsZS
5jb20vVXNlcnMvNDRmNjE0MmRmOTZiZDZhYjYxZTc1MjFkOSIsImF0dHJpYnV0ZXM
iOlsiaWQiLCJuYW1lIiwidXNlck5hbWUiLCJwYXNzd29yZCIsImVtYWlscyJdfX19
.
```

Figure 6: Example Unsecured Security Event Token

For the purpose of having a simpler example in Figure 6, an unsecured token is shown. When SETs are not signed or encrypted, other mechanisms such as TLS MUST be employed to provide integrity protection, confidentiality, and issuer authenticity, 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. Requirements for SET Profiles

Profiling specifications of this specification define actual SETs to be used in particular use cases. These profiling specifications define the syntax and semantics of SETs conforming to that SET profile and rules for validating those SETs. Profiling specifications SHOULD define syntax, semantics, subject identification, and validation.

Syntax
: The syntax of the SETs defined, including:

  Top-Level Claims
  : Claims and values in the JWT Claims Set. Examples are claims defined by the JWT specification [RFC7519], this specification, and by the profiling specification.

  Event Payload
  : The JSON data structure contents and format, containing event-specific information, if any (see Section 1.2).

Semantics
: 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.

Subject Identification
: 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 in cases in which the subject is not globally unique and has a different issuer from the SET itself.

Validation
: 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.

Among the syntax and semantics of SETs that a profiling specification may define is whether the value of the "events" claim may contain multiple members, and what processing instructions are employed in the single- and multiple-valued cases for SETs conforming to that profile. Many valid choices are possible. For instance, some profiles might allow multiple event identifiers to be present and specify that any that are not understood by recipients be ignored, thus enabling extensibility. Other profiles might allow multiple event identifiers to be present but require that all be understood if the SET is to be accepted. Some profiles might require that only a single value be present. All such choices are within the scope of profiling specifications to define.

4. Preventing Confusion between SETs and Other JWTs

Because [RFC7519] states that "all claims that are not understood by implementations MUST be ignored", there is a consideration that a SET might be confused with another kind of JWT from the same issuer. Unless this confusion is prevented, this might enable an attacker who possesses a SET to use it in a context in which another kind of JWT is expected, or vice versa. This section presents concrete techniques for preventing confusion between SETs and several other specific kinds of JWTs, as well as generic techniques for preventing possible confusion between SETs and other kinds of JWTs.

4.1. Distinguishing SETs from ID Tokens

A SET might be confused with an ID Token [OpenID.Core] if a SET is mistakenly or maliciously 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 2.3, will not achieve disambiguation between ID Tokens and SETs, as the ID Token validation rules do not use the "typ" header parameter value.

4.2. 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, also implementation specific, generally. 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 SET recipient 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 2.3, and modify access token validation systems to use the "typ" header parameter value.

4.3. 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 Parameter to Indicate Received Realm" [RFC8055] and "PASSporT: Personal Assertion Token" [RFC8225] 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 2.3, 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. Validating that the JWT has an "events" claim will be effective in preventing attackers from passing other kinds of JWTs off as SETs.

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 [RFC8225], 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" [JWT-BCP]. The proposed best practices in that document may also be applicable for particular SET profiles and use cases.

5. Security Considerations

5.1. Confidentiality and Integrity

SETs may 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] or a higher version and MAY support additional transport-layer mechanisms meeting its security requirements. When using TLS, the client MUST perform a TLS server certificate check, per [RFC6125]. Implementation security considerations for TLS can be found in "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)" [RFC7525].

Security events distributed through third parties or that carry personally identifiable information MUST 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] by an issuer that is trusted to do so for the use case so that the SET can be authenticated and validated by the SET recipient.

5.2. Delivery

This specification does not define a delivery mechanism for SETs. 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 (mutual) TLS, there is no assurance that the correct endpoint received the SET and that it could be successfully processed.

5.3. Sequencing

This specification defines no means of ordering multiple SETs in a sequence. Depending on the type and nature of the events represented by SETs, order may or may not matter. For example, in provisioning, event order is critical -- an object cannot be modified before it is created. In other SET types, such as a token revocation, the order of SETs for revoked tokens does not matter. If, however, the event conveys a logged 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. Thus, time by itself will not guarantee order.

Specifications profiling SET SHOULD define a mechanism for detecting order or sequence of events when the order matters. For example, the "txn" claim could contain an ordered value (e.g., a counter) that the issuer includes, although just as for timestamps, ensuring such ordering can be difficult in distributed systems.

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 a SET recipient in advance of or behind the process that caused the event. For example, a user having been required to log out and then log back in again, may cause a "token revoked" SET to be issued, typically causing the receiver to reset all active sessions at the receiver that are related to that user. If a revocation SET arrives at the

same time as the user agent re-logs in, timing could cause problems by erroneously treating the new user session as logged out. Profiling specifications SHOULD be careful to consider both SET expression and timing issues. For example, it might be more appropriate to revoke a specific session or ID Token rather than a general logout statement about a "user". Alternatively, profiling specifications could use timestamps that allow new sessions to be started immediately after a stated logout event time.

5.5. Preventing Confusion

Also, see Section 4 above for both additional security considerations and normative text on preventing SETs from being confused with other kinds of JWTs.

6. Privacy Considerations

If a SET needs to be retained for audit purposes, the signature can be used to provide verification of its authenticity.

SET issuers SHOULD attempt to specialize SETs so that their content is targeted to the specific business and protocol needs of the intended SET recipients.

When sharing personally identifiable information or information that is otherwise considered confidential to affected users, SET issuers and recipients should have the appropriate legal agreements and user consent and/or terms of service in place.

The propagation of subject identifiers can be perceived as personally identifiable information. Where possible, SET issuers and recipients SHOULD devise approaches that prevent propagation -- for example, the passing of a salted hash value that requires the SET recipient to know the subject.

In some cases, it may be possible for a SET recipient to correlate different events and thereby gain information about a subject that the SET issuer did not intend to share. For example, a SET recipient might be able to use "iat" values or highly precise "toe" values to determine that two otherwise un-relatable events actually relate to the same real-world event. The union of information from both events could allow a SET recipient to de-anonymize data or recognize that unrelated identifiers relate to the same individual. SET issuers SHOULD take steps to minimize the chance of event correlation, when such correlation would constitute a privacy violation. For instance, they could use approximate values for the "toe" claim or arbitrarily delay SET issuance, where such delay can be tolerated.

7. IANA Considerations

7.1. JSON Web Token Claims Registration

IANA has registered the "events", "toe", and "txn" claims in the IANA "JSON Web Token Claims" registry [IANA.JWT.Claims] established by [RFC7519].

7.1.1. Registry Contents

- Claim Name: "events"
- Claim Description: Security Events
- Change Controller: IESG
- Specification Document(s): Section 2.2 of [RFC8417]

- Claim Name: "toe"
- Claim Description: Time of Event
- Change Controller: IESG
- Specification Document(s): Section 2.2 of [RFC8417]

- Claim Name: "txn"
- Claim Description: Transaction Identifier
- Change Controller: IESG
- Specification Document(s): Section 2.2 of [RFC8417]

7.2. Structured Syntax Suffix Registration

IANA has registered the "+jwt" structured syntax suffix [RFC6838] in the "Structured Syntax Suffix" registry [IANA.StructuredSuffix] in the manner described in [RFC6838], which can be used to indicate that the media type is encoded as a JWT.

7.2.1. Registry Contents

- Name: JSON Web Token (JWT)
- +suffix: +jwt
- References: Section 3 of [RFC7519], Section 7.2 of [RFC8417]
- Encoding Considerations: binary; JWT values are encoded as a series of base64url-encoded values (with trailing '=' characters removed), some of which may be the empty string, separated by period ('.') characters.
- Interoperability Considerations: N/A
- Fragment Identifier Considerations:
  The syntax and semantics of fragment identifiers specified for +jwt SHOULD be as specified for "application/jwt". (At publication of this document, there is no fragment identification syntax defined for "application/jwt".)

  The syntax and semantics for fragment identifiers for a specific "xxx/yyy+jwt" SHOULD be processed as follows:

  For cases defined in +jwt where the fragment identifier resolves per the +jwt rules, process as specified in +jwt.

  For cases defined in +jwt where the fragment identifier does not resolve per the +jwt rules, process as specified in "xxx/yyy+jwt".

  For cases not defined in +jwt, process as specified in "xxx/yyy+jwt".
- Security Considerations: See Section 11 of [RFC7519].
- Contact:
  Michael B. Jones, mbj@microsoft.com
- Author/Change Controller:
  Security Events Working Group.
  The IESG has change control over this registration.

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: binary; A SET is a JWT; JWT values are encoded as a series of base64url-encoded values (with trailing '=' characters removed), some of which may be the empty string, separated by period ('.') characters.
- Security considerations: See Section 5 of [RFC8417]
- Interoperability considerations: N/A
- Published specification: Section 2.3 of [RFC8417]
- Applications that use this media type: Applications that exchange SETs
- 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

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

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

[IANA.StructuredSuffix]
IANA, "Structured Syntax Suffix",
<https://www.iana.org/assignments/
media-type-structured-suffix/>.

[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>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <https://www.rfc-editor.org/info/rfc3629>.

[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>.

[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>.

[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>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

8.2. Informative References

[JWT-BCP] Sheffer, Y., Hardt, D., and M. Jones, "JSON Web Token Best Current Practices", Work in Progress, draft-ietf-oauth-jwt-bcp-03, May 2018.

[OpenID.BackChannel] Jones, M. and J. Bradley, "OpenID Connect Back-Channel Logout 1.0", January 2017, <http://openid.net/specs/openid-connect-backchannel-1_0.html>.

[OpenID.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>.

[OpenID.RISC.Events] Scurtescu, M., Backman, A., Hunt, P., and J. Bradley, "OpenID RISC Event Types 1.0", April 2018, <http://openid.net/specs/openid-risc-event-types-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>.

[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, <https://www.rfc-editor.org/info/rfc7644>.

[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>.

[RFC8225] Wendt, C. and J. Peterson, "PASSporT: Personal Assertion Token", RFC 8225, DOI 10.17487/RFC8225, February 2018, <https://www.rfc-editor.org/info/rfc8225>.

Acknowledgments

The editors would like to thank the members of the IETF SCIM working group, 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, the Security Events working group, and related working groups for their contributions to this specification. The specification incorporates suggestions made by many people, including Annabelle Backman, John Bradley, Alissa Cooper, Ned Freed, Dick Hardt, Russ Housley, Benjamin Kaduk, Mirja Kuehlewind, Mark Lizar, Alexey Melnikov, Andrew Nash, Eric Rescorla, Adam Roach, Justin Richer, Nat Sakimura, Marius Scurtescu, Yaron Sheffer, and Martin Vigoureux.

Authors' Addresses

Phil Hunt (editor)
Oracle Corporation

Email: phil.hunt@yahoo.com

Michael B. Jones
Microsoft

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

William Denniss
Google

Email: rfc8417@wdenniss.com
URI: https://wdenniss.com/SET

Morteza Ansari
Cisco

Email: morteza.ansari@cisco.com