Security Events Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 28, 2020

A. Backman, Ed.
Amazon
M. Jones, Ed.
Microsoft
M. Scurtescu
Coinbase
M. Ansari
Cisco
A. Nadalin
Microsoft

June 26, 2020

Push-Based Security Event Token (SET) Delivery Using HTTP
draft-ietf-secevent-http-push-14

Abstract

This specification defines how a Security Event Token (SET) can be delivered to an intended recipient using HTTP POST over TLS. The SET is transmitted in the body of an HTTP POST request to an endpoint operated by the recipient, and the recipient indicates successful or failed transmission via the HTTP response.

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 December 28, 2020.

Table of Contents

1. Introduction and Overview

This specification defines a mechanism by which a transmitter of a Security Event Token (SET) [RFC8417] can deliver the SET to an intended SET Recipient via HTTP POST [RFC7231] over TLS. This is an alternative SET delivery method to the one defined in [I-D.ietf-secevent-http-poll].

Push-based SET delivery over HTTP POST is intended for scenarios where all of the following apply:

- The transmitter of the SET is capable of making outbound HTTP requests.
- The recipient is capable of hosting a TLS-enabled HTTP endpoint that is accessible to the transmitter.
- The transmitter and recipient are willing to exchange data with one another.

In some scenarios, either push-based or poll-based delivery could be used, and in others, only one of them would be applicable.

A mechanism for exchanging configuration metadata such as endpoint URLs, cryptographic keys, and possible implementation constraints such as buffer size limitations between the transmitter and recipient is out of scope for this specification. How SETs are defined and the process by which security events are identified for SET Recipients are specified in [RFC8417].

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.

Throughout this document, all figures may contain spaces and extra line wrapping for readability and due to space limitations.

1.2. Definitions

This specification utilizes the following terms defined in [RFC8417]: "Security Event Token (SET)", "SET Issuer", "SET Recipient", and "Event Payload", as well as the term defined below:

SET Transmitter An entity that delivers SETs in its possession to one or more SET Recipients.

2. SET Delivery

To deliver a SET to a given SET Recipient, the SET Transmitter makes a SET transmission request to the SET Recipient, with the SET itself contained within the request. The SET Recipient replies to this request with a response either acknowledging successful transmission

of the SET or indicating that an error occurred while receiving, parsing, and/or validating the SET.

Upon receipt of a SET, the SET Recipient SHALL validate that all of the following are true:

- The SET Recipient can parse the SET.
- The SET is authentic (i.e., it was issued by the issuer specified within the SET, and if signed, was signed by a key belonging to the issuer).
- The SET Recipient is identified as an intended audience of the SET.
- The SET Issuer is recognized as an issuer that the SET Recipient is willing to receive SETs from (e.g., the issuer is listed as allowed by the SET Recipient).
- The SET Recipient is willing to accept this SET from this SET Transmitter (e.g., the SET Transmitter is expected to send SETs with the issuer and subject of the SET in question).

The mechanisms by which the SET Recipient performs this validation are out of scope for this document. SET parsing, issuer identification, and audience identification are defined in [RFC8417]. The mechanism for validating the authenticity of a SET is deployment specific, and may vary depending on the authentication mechanisms in use, and whether the SET is signed and/or encrypted (See Section 3).

SET Transmitters MAY transmit SETs issued by another entity. The SET Recipient may accept or reject (i.e., return an error response such as "access_denied") a SET at its own discretion.

The SET Recipient persists the SET in a way that is sufficient to meet the SET Recipient’s own reliability requirements. The level and method of retention of SETs by SET Recipients is out of scope of this specification. Once the SET has been validated and persisted, the SET Recipient SHOULD immediately return a response indicating that the SET was successfully delivered. The SET Recipient SHOULD NOT perform further processing of the SET beyond the required validation steps prior to sending this response. Any additional steps SHOULD be executed asynchronously from delivery to minimize the time the SET Transmitter is waiting for a response.

The SET Transmitter MAY transmit the same SET to the SET Recipient multiple times, regardless of the response from the SET Recipient. The SET Recipient MUST respond as it would if the SET had not been

previously received by the SET Recipient. The SET Recipient MUST NOT expect or depend on a SET Transmitter to re-transmit a SET or otherwise make a SET available to the SET Recipient once the SET Recipient acknowledges that it was received successfully.

The SET Transmitter should not re-transmit a SET unless the SET Transmitter suspects that previous transmissions may have failed due to potentially recoverable errors (such as network outage or temporary service interruption at either the SET Transmitter or SET Recipient). In all other cases, the SET Transmitter SHOULD NOT re-transmit a SET. The SET Transmitter SHOULD delay retransmission for an appropriate amount of time to avoid overwhelming the SET Recipient (see Section 4).

2.1. Transmitting a SET

To transmit a SET to a SET Recipient, the SET Transmitter makes an HTTP POST request to a TLS-enabled HTTP endpoint provided by the SET Recipient. The "Content-Type" header field of this request MUST be "application/secevent+jwt" as defined in Sections 2.3 and 7.2 of [RFC8417], and the "Accept" header field MUST be "application/json". The request body MUST consist of the SET itself, represented as a JWT [RFC7519].

The SET Transmitter MAY include in the request an "Accept-Language" header field to indicate to the SET Recipient the preferred language(s) in which to receive error messages.

The mechanisms by which the SET Transmitter determines the HTTP endpoint to use when transmitting a SET to a given SET Recipient are not defined by this specification and are deployment specific.

The following is a non-normative example of a SET transmission request:

```
POST /Events HTTP/1.1
Host: notify.rp.example.com
Accept: application/json
Accept-Language: en-US, en;q=0.5
Content-Type: application/secevent+jwt

eyJ0eXAiOiJzZWNldmVudCtqd3QiLCJhbGciOiJIUzI1NiJ9Cg
.
eyJpc3MiOiJodHRwczovL2lkcC5leGFtcGxlLmNvbS8iLCJqdGkiOiI3NTZFNjk
3MTc1NjUyMDY5NjQ2NTZFNzQ2OTY2Njk2NTcyIiwiaWF0IjoxNTA4MTg0ODQ1LC
JhdWQiOiI2MzZDNjk2NTZFNzQ1RjY5NjQiLCJldmVudHMiOnsiaHR0cHM6Ly9zY
2hlbWFzLm9wZW5pZC5uZXQvc2VjZXZlbnQvcmlzYy9ldmVudC10eXBlL2FjY291
bnQtZGlzYWJsZWQiOnsic3ViamVjdCI6eyJzdWJqZWN0X3R5cGUiOiJpc3Mtc3V
iIiwiaXNzIjoiaHR0cHM6Ly9pZHAuZXhhbXBsZS5jb20vIiwic3ViIjoiNzM3NT
YyNkE2NTYzNzQifSwicmVhc29uIjoiaGlqYWNraW5nIn19fQ
.
Y4rXxMD406P2edv00cr9Wf3_XwNtLjB9n-jTqN1_lLc
```

Figure 1: Example SET Transmission Request

2.2. Success Response

If the SET is determined to be valid, the SET Recipient SHALL acknowledge successful transmission by responding with HTTP Response Status Code 202 (Accepted) (see Section 6.3.3 of [RFC7231]). The body of the response MUST be empty.

The following is a non-normative example of a successful receipt of a SET.

```
HTTP/1.1 202 Accepted
```

Figure 2: Example Successful Delivery Response

2.3. Failure Response

In the event of a general HTTP error condition, the SET Recipient responds with the applicable HTTP Status Code, as defined in Section 6 of [RFC7231].

When the SET Recipient detects an error parsing, validating, or authenticating a SET transmitted in a SET Transmission Request, the SET Recipient SHALL respond with an HTTP Response Status Code of 400 (Bad Request). The "Content-Type" header field of this response MUST

be "application/json", and the body MUST be a UTF-8 encoded JSON [RFC8259] object containing the following name/value pairs:

err A Security Event Token Error Code (see Section 2.4).

description A UTF-8 string containing a human-readable description of the error that may provide additional diagnostic information. The exact content of this field is implementation specific.

The response MUST include a "Content-Language" header field, whose value indicates the language of the error descriptions included in the response body. If the SET Recipient can provide error descriptions in multiple languages, they SHOULD choose the language to use according to the value of the "Accept-Language" header field sent by the SET Transmitter in the transmission request, as described in Section 5.3.5 of [RFC7231]. If the SET Transmitter did not send an "Accept-Language" header field, or if the SET Recipient does not support any of the languages included in the header field, the SET Recipient MUST respond with messages that are understandable by an English-speaking person, as described in Section 4.5 of [RFC2277].

The following is a non-normative example error response indicating that the key used to encrypt the SET has been revoked.

```
HTTP/1.1 400 Bad Request
Content-Language: en-US
Content-Type: application/json

{
  "err": "invalid_key",
  "description": "Key ID 12345 has been revoked."
}
```

Figure 3: Example Error Response (invalid_key)

The following is a non-normative example error response indicating that the access token included in the request is expired.

```
HTTP/1.1 400 Bad Request
Content-Language: en-US
Content-Type: application/json

{
  "err": "authentication_failed",
  "description": "Access token has expired."
}
```

Figure 4: Example Error Response (authentication_failed)

The following is a non-normative example error response indicating that the SET Receiver is not willing to accept SETs issued by the specified issuer from this particular SET Transmitter.

```
HTTP/1.1 400 Bad Request
Content-Language: en-US
Content-Type: application/json

{
  "err": "invalid_issuer",
  "description": "Not authorized for issuer https://iss.example.com/."
}
```

Figure 5: Example Error Response (access_denied)

2.4. Security Event Token Delivery Error Codes

Security Event Token Delivery Error Codes are strings that identify a specific category of error that may occur when parsing or validating a SET. Every Security Event Token Delivery Error Code MUST have a unique name registered in the IANA "Security Event Token Delivery Error Codes" registry established by Section 7.1.

The following table presents the initial set of Error Codes that are registered in the IANA "Security Event Token Delivery Error Codes" registry:

Error Code	Description
invalid_request	The request body cannot be parsed as a SET, or the Event Payload within the SET does not conform to the event's definition.
invalid_key	One or more keys used to encrypt or sign the SET is invalid or otherwise unacceptable to the SET Recipient (expired, revoked, failed certificate validation, etc.).
invalid_issuer	The SET issuer is invalid for the SET Recipient.
invalid_audience	The SET audience does not correspond to the SET Recipient.
authentication_failed	The SET Recipient could not authenticate the SET Transmitter.
access_denied	The SET Transmitter is not authorized to transmit the SET to the SET Recipient.

Table 1: SET Delivery Error Codes

Other Error Codes may also be received, as the set of Error Codes is extensible via the IANA "Security Event Token Delivery Error Codes" registry established in Section 7.1.

3. Authentication and Authorization

The SET delivery method described in this specification is based upon HTTP over TLS [RFC2818] and standard HTTP authentication and authorization schemes, as per [RFC7235]. The TLS server certificate MUST be validated using DNS-ID [RFC6125] and/or DANE [RFC6698].

Authorization for the eligibility to provide actionable SETs can be determined by using the identity of the SET Issuer, the identity of the SET Transmitter, perhaps using mutual TLS, or via other employed authentication methods. Because SETs are not commands, SET Recipients are free to ignore SETs that are not of interest.

4. Delivery Reliability

Delivery reliability requirements may vary depending upon the use cases. This specification defines the response from the SET Recipient in such a way as to provide the SET Transmitter with the information necessary to determine what further action is required, if any, in order to meet their requirements. SET Transmitters with

high reliability requirements may be tempted to always retry failed transmissions. However, it should be noted that for many types of SET delivery errors, a retry is extremely unlikely to be successful. For example, "invalid_request" indicates a structural error in the content of the request body that is likely to remain when re-transmitting the same SET. Others such as "access_denied" may be transient, for example if the SET Transmitter refreshes expired credentials prior to re-transmission.

The SET Transmitter may be unaware of whether or not a SET has been delivered to a SET Recipient. For example, a network interruption could prevent the SET Transmitter from receiving the success response, or a service outage could prevent the SET Transmitter from recording the fact that the SET was delivered. It is left to the implementer to decide how to handle such cases, based on their requirements. For example, it may be appropriate for the SET Transmitter to re-transmit the SET to the SET Recipient, erring on the side of guaranteeing delivery, or it may be appropriate to assume delivery was successful, erring on the side of not spending resources re-transmitting previously delivered SETs. Other options, such as sending the SET to a "dead letter queue" for manual examination may also be appropriate.

Implementers SHOULD evaluate the reliability requirements of their use cases and the impact of various retry mechanisms and re-transmission policies on the performance of their systems to determine an appropriate strategy for handling various error conditions.

5. Security Considerations

5.1. Authentication Using Signed SETs

JWS signed SETs can be used (see [RFC7515] and Section 5 of [RFC8417]) to enable the SET Recipient to validate that the SET Issuer is authorized to provide actionable SETs.

5.2. HTTP Considerations

SET delivery depends on the use of Hypertext Transfer Protocol and is thus subject to the security considerations of HTTP Section 9 of [RFC7230] and its related specifications.

5.3. Confidentiality of SETs

SETs may contain sensitive information, including Personally Identifiable Information (PII), or be distributed through third parties. In such cases, SET Transmitters and SET Recipients MUST

protect the confidentiality of the SET contents. TLS MUST be used to secure the transmitted SETs. In some use cases, encrypting the SET as described in JWE [RFC7516] will also be required. The Event delivery endpoint MUST support at least TLS version 1.2 [RFC5246] and SHOULD support the newest version of TLS that meets its security requirements, which as of the time of this publication is TLS 1.3 [RFC8446]. The client MUST perform a TLS/SSL server certificate check using DNS-ID [RFC6125] and/or DANE [RFC6698]. How a SET Transmitter determines the expected service identity to match the SET Recipient's server certificate against is out of scope for this document. The implementation security considerations for TLS in "Recommendations for Secure Use of TLS and DTLS" [RFC7525] MUST be followed.

5.4. Denial of Service

The SET Recipient may be vulnerable to a denial-of-service attack where a malicious party makes a high volume of requests containing invalid SETs, causing the endpoint to expend significant resources on cryptographic operations that are bound to fail. This may be mitigated by authenticating SET Transmitters with a mechanism such as mutual TLS. Rate-limiting problematic transmitters is also a possible means of mitigation.

5.5. Authenticating Persisted SETs

At the time of receipt, the SET Recipient can rely upon TLS mechanisms, HTTP authentication methods, and/or other context from the transmission request to authenticate the SET Transmitter and validate the authenticity of the SET. However, this context is typically unavailable to systems to which the SET Recipient forwards the SET, or to systems that retrieve the SET from storage. If the SET Recipient requires the ability to validate SET authenticity outside of the context of the transmission request, then the SET Recipient SHOULD ensure that such SETs have been signed in accordance with [RFC7515]. Needed context could also be stored with the SET and retrieved with it.

6. Privacy Considerations

SET Transmitters should attempt to deliver SETs that are 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, SET Transmitters and Recipients MUST have the appropriate legal agreements and user consent or terms of service in place. Furthermore, data that needs confidentiality protection MUST be

encrypted, at least with TLS and sometimes also using JSON Web Encryption (JWE) [RFC7516].

In some cases, subject identifiers themselves may be considered sensitive information, such that their inclusion within a SET may be considered a violation of privacy. SET Issuers and SET Transmitters should consider the ramifications of sharing a particular subject identifier with a SET Recipient (e.g., whether doing so could enable correlation and/or de-anonymization of data) and choose appropriate subject identifiers for their use cases.

7. IANA Considerations

7.1. Security Event Token Delivery Error Codes

This document defines Security Event Token Delivery Error Codes, for which IANA is asked to create and maintain a new registry titled "Security Event Token Delivery Error Codes". Initial values for the Security Event Token Delivery Error Codes registry are defined in Table 1 and registered below. Future assignments are to be made through the Specification Required registration policy ([RFC8126]) and shall follow the template below.

Error Codes are intended to be interpreted by automated systems, and therefore SHOULD identify classes of errors to which an automated system could respond in a meaningfully distinct way (e.g., by refreshing authentication credentials and retrying the request).

Error Code names are case sensitive. Names may not match other registered names in a case-insensitive manner unless the Designated Experts state that there is a compelling reason to allow an exception.

Criteria that should be applied by the Designated Experts includes determining whether the proposed registration duplicates existing functionality, whether it is likely to be of general applicability or whether it is useful only for a single application, and whether the registration description is clear.

It is suggested that multiple Designated Experts be appointed who are able to represent the perspectives of different applications using this specification, in order to enable broadly informed review of registration decisions. In cases where a registration decision could be perceived as creating a conflict of interest for a particular Expert, that Expert should defer to the judgment of the other Experts.

7.1.1. Registration Template

Error Code
: The name of the Security Event Token Delivery Error Code, as described in Section 2.4. The name MUST be a case-sensitive ASCII string consisting only of letters, digits, and underscore; these are the characters whose codes fall within the inclusive ranges 0x30-39, 0x41-5A, 0x5F and 0x61-7A.

Description
: A brief human-readable description of the Security Event Token Delivery Error Code.

Change Controller
: For error codes registered by the IETF or its working groups, list "IETF". For all other error codes, list the name of the party responsible for the registration. Contact information such as mailing address, email address, or phone number may also be provided.

Defining Document(s)
: A reference to the document or documents that define the Security Event Token Delivery Error Code. The definition MUST specify the name and description of the error code and explain under what circumstances the error code may be used. URIs that can be used to retrieve copies of each document at no cost SHOULD be included.

7.1.2. Initial Registry Contents

Error Code: invalid_request
Description: The request body cannot be parsed as a SET or the event payload within the SET does not conform to the event's definition.
Change Controller: IETF
Defining Document(s): Section 2.4 of [[this specification]]

Error Code: invalid_key
Description: One or more keys used to encrypt or sign the SET is invalid or otherwise unacceptable to the SET Recipient (expired, revoked, failed certificate validation, etc.).
Change Controller: IETF
Defining Document(s): Section 2.4 of [[this specification]]

Error Code: invalid_issuer
Description: The SET issuer is invalid for the SET Recipient.
Change Controller: IETF
Defining Document(s): Section 2.4 of [[this specification]]

Error Code: invalid_audience
Description: The SET audience does not correspond to the SET Recipient.
Change Controller: IETF
Defining Document(s): Section 2.4 of [[this specification]]

Error Code: authentication_failed
Description: The SET Recipient could not authenticate the SET Transmitter.
Change Controller: IETF
Defining Document(s): Section 2.4 of [[this specification]]

Error Code: access_denied
Description: The SET Transmitter is not authorized to transmit the SET to the SET Recipient.
Change Controller: IETF
Defining Document(s): Section 2.4 of [[this specification]]

8. References

8.1. Normative References

[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>.

[RFC2277] Alvestrand, H., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, DOI 10.17487/RFC2277, January 1998, <https://www.rfc-editor.org/info/rfc2277>.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <https://www.rfc-editor.org/info/rfc2818>.

[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>.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <https://www.rfc-editor.org/info/rfc6698>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <https://www.rfc-editor.org/info/rfc7231>.

[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>.

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

[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>.

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

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

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

8.2. Informative References

[I-D.ietf-secevent-http-poll] Backman, A., Jones, M., Scurtescu, M., Ansari, M., and A. Nadalin, "Poll-Based Security Event Token (SET) Delivery Using HTTP", draft-ietf-secevent-http-poll-12 (work in progress), June 2020.

[RFC7235] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Authentication", RFC 7235, DOI 10.17487/RFC7235, June 2014, <https://www.rfc-editor.org/info/rfc7235>.

Appendix A. Unencrypted Transport Considerations

Earlier versions of this specification made the use of TLS optional and described security and privacy considerations resulting from use of unencrypted HTTP as the underlying transport. When the working group decided to mandate usage HTTP over TLS, it also decided to preserve the description of these considerations in this non-normative appendix.

SETs may contain sensitive information that is considered Personally Identifiable Information (PII). In such cases, SET Transmitters and SET Recipients MUST protect the confidentiality of the SET contents. When TLS is not used, this means that the SET MUST be encrypted as described in JWE [RFC7516].

If SETs were allowed to be transmitted over unencrypted channels, some privacy-sensitive information about them might leak, even though the SETs themselves are encrypted. For instance, an attacker may be able to determine whether or not a SET was accepted and the reason for its rejection or may be able to derive information from being able to observe the size of the encrypted SET. (Note that even when TLS is utilized, some information leakage is still possible; message padding algorithms to prevent side channels remain an open research topic.)

Appendix B. Other Streaming Specifications

[[NOTE TO THE RFC EDITOR: This section to be removed prior to publication]]

The following pub/sub, queuing, and streaming systems were reviewed as possible solutions or as input to the current draft:

Poll-Based Security Event Token (SET) Delivery Using HTTP

In addition to this specification, the WG is defining a polling-based SET delivery protocol. That protocol [I-D.ietf-secevent-http-poll] describes it as:

This specification defines how a series of Security Event Tokens (SETs) can be delivered to an intended recipient using HTTP POST over TLS initiated as a poll by the recipient. The specification also defines how delivery can be assured, subject to the SET Recipient's need for assurance.

XMPP Events

The WG considered XMPP Events and their ability to provide a single messaging solution without the need for both polling and push modes. The feeling was the size and methodology of XMPP was too far apart from the current capabilities of the SECEVENTs community, which focuses in on HTTP based service delivery and authorization.

Amazon Simple Notification Service

Simple Notification Service is a pub/sub messaging product from AWS. SNS supports a variety of subscriber types: HTTP/HTTPS endpoints, AWS Lambda functions, email addresses (as JSON or plain text), phone numbers (via SMS), and AWS SQS standard queues. It does not directly support pull, but subscribers can get the pull model by creating an SQS queue and subscribing it to the topic. Note that this puts the cost of pull support back onto the subscriber, just as it is in the push model. It is not clear that one way is strictly better than the other; larger, sophisticated developers may be happy to own message persistence so they can have their own internal delivery guarantees. The long tail of OIDC clients may not care about that or may fail to get it right. Regardless, I think we can learn something from the Delivery Policies supported by SNS, as well as the delivery controls that SQS offers (e.g., Visibility Timeout, Dead-Letter Queues). I am not suggesting that we need all of these things in the spec, but they give an idea of what features people have found useful.

Other information:

- API Reference:
  http://docs.aws.amazon.com/AWSSimpleQueueService/latest/APIReference/Welcome.html

- Visibility Timeouts:
  http://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-visibility-timeout.html

Apache Kafka

Apache Kafka is an Apache open source project based upon TCP for distributed streaming. It prescribes some interesting general-purpose features that seem to extend far beyond the simpler streaming model that SECEVENTs is after. A comment from MS has been that Kafka does an acknowledge with poll combination event which seems to be a performance advantage. See: https://kafka.apache.org/intro

Google Pub/Sub

The Google Pub Sub system favors a model whereby polling and acknowledgement of events is done with separate endpoints and as separate functions.

Information:

- Cloud Overview - https://cloud.google.com/pubsub/

- Subscriber Overview - https://cloud.google.com/pubsub/docs/subscriber

- Subscriber Pull(poll) - https://cloud.google.com/pubsub/docs/pull

Appendix C. Acknowledgments

The editors would like to thank the members of the SCIM working group, which began discussions of provisioning events starting with draft-hunt-scim-notify-00 in 2015. We would like to thank Phil Hunt and the other authors of draft-ietf-secevent-delivery-02, upon which this specification is based. We would like to thank the participants in the SecEvents working group for their contributions to this specification.

Additionally, we would like to thank the following individuals for their reviews of the specification: Joe Clarke, Roman Danyliw, Vijay Gurbani, Benjamin Kaduk, Erik Kline, Murray Kucherawy, Barry Leiba, Yaron Sheffer, Robert Sparks, Valery Smyslov, Eric Vyncke, and Robert Wilton.

Appendix D. Change Log

[[NOTE TO THE RFC EDITOR: This section to be removed prior to publication]]

Draft 00 - AB - Based on draft-ietf-secevent-delivery-02 with the following changes:

- Renamed to "Push-Based SET Token Delivery Using HTTP"
- Removed references to the HTTP Polling delivery method.
- Removed informative reference to RFC6202.

Draft 01 - AB:

- Fixed area and workgroup to match secevent.
- Removed unused definitions and definitions already covered by SET.
- Renamed Event Transmitter and Event Receiver to SET Transmitter and SET Receiver, respectively.
- Added IANA registry for SET Delivery Error Codes.
- Removed enumeration of HTTP authentication methods.
- Removed generally applicable guidance for HTTP, authorization tokens, and bearer tokens.
- Moved guidance for using authentication methods as DoS protection to Security Considerations.
- Removed redundant instruction to use WWW-Authenticate header.
- Removed further generally applicable guidance for authorization tokens.
- Removed bearer token from example delivery request, and text referencing it.
- Broke delivery method description into separate request/response sections.
- Added missing empty line between headers and body in example request.
- Removed inapplicable notes about example formatting.

- Removed text about SET creation and handling.
- Removed duplication in protocol description.
- Added "non-normative example" text to example transmission request.
- Fixed inconsistencies in use of Error Code term.

Draft 02 - AB:

- Rewrote abstract and introduction.
- Rewrote definitions for SET Transmitter, SET Receiver.
- Renamed Event Delivery section to SET Delivery.
- Readability edits to Success Response and Failure Response sections.
- Consolidated definition of error response under Failure Response section.
- Removed Event Delivery Process section and moved its content to parent section.
- Readability edits to SET Delivery section and its subsections.
- Added callout that SET Receiver HTTP endpoint configuration is out-of-scope.
- Added callout that SET verification mechanisms are out-of-scope.
- Added retry guidance, notes regarding delivery reliability requirements.
- Added guidance around using JWS and/or JWE to authenticate persisted SETs.

Draft 03 - mbj:

- Addressed problems identified in my 18-Jul-18 review message titled "Issues for both the Push and Poll Specs".
- Changes to align terminology with RFC 8417, for instance, by using the already defined term SET Recipient rather than SET Receiver.
- Applied editorial and minor normative corrections.

- Updated Marius' contact information.

Draft 04 - AB:

- Replaced Error Codes with smaller set of meaningfully differentiated codes.
- Added more error response examples.
- Removed un-referenced normative references.
- Added normative reference to JSON in error response definition.
- Added text clarifying that the value of the "description" attribute in error responses is implementation specific.
- Added requirement that error descriptions and responses are UTF-8 encoded.
- Added error description language preferences and specification via "Accept-Language" and "Content-Language" headers.
- Added "recognized issuer" validation requirement in section 2.
- Added timeouts as an acceptable reason to resend a SET in section 2.
- Edited text in section 1 to clarify that configuration is out of scope.
- Made minor editorial corrections.

Draft 05 - AB:

- Made minor editorial corrections.
- Updated example request with a correct SET header and signature.
- Revised TLS guidance to allow implementers to provide confidentiality protection via JWE.
- Revised TLS guidance to require *at least* TLS 1.2.
- Revised TLS guidance to recommend supporting the newest version of TLS that meets security requirements.
- Revised SET Delivery Error Code format to allow the same set of characters as is allowed in error codes in RFC6749.

- Added mention of HTTP Poll spec to list of other streaming specs in appendix.
- Added validation step requiring SET Recipient to verify that the SET is one which the SET Transmitter is expected to send to the SET Recipient.
- Changed responding to errors with an appropriate HTTP status code from optional to recommended.
- Changed Error Codes registry change policy from Expert Review to First Come First Served; added guidance that error codes are meant to be consumed by automated systems.
- Added text making clear that it is up to SET Recipients whether or not they will accept SETs where the SET Issuer is different from the SET Transmitter.
- Reworded guidance around signing and/or encrypting SETs for integrity protection.
- Renamed TLS "Support Considerations" section to "Confidentiality of SETs".
- Reworded guidance around subject identifier selection and privacy concerns.

Draft 06 - mbj, MS:

- Made minor editorial corrections.
- Updated to indicate that failure response should be returned if errors occur in authenticating the SET.
- Updated reference for JSON from RFC 7159 to RFC 8259.
- Fixed Authentication Using Signed SETs to indicate the SET Transmitter must be authorized to deliver the SET, not the SET Issuer.
- Fixed Authenticating Persisted SETs to put the responsibility for ensuring the SET is signed on the SET Recipient.
- Fixed error code format definition to match error codes defined in doc.

Draft 07 - AB:

- Made minor editorial corrections.
- Removed "SET Recipient" definition and added explicit list of terms used from RFC8417.

Draft 08 - mbj

- Addressed area director review comments by Benjamin Kaduk.

Draft 09 - mbj + AB

- Corrected editorial nits.

Draft 10 - AB

- Addressed area director review comments by Benjamin Kaduk:
  - Added reference to 8417 as definition document for SETs.
  - Added text clarifying that determining the SET Recipient's service identity is out of scope.
  - Added normative recommendation for transmitters to target SETs to specific business needs of subscribers.
  - Minor editorial corrections.

Draft 11 - mbj

- Addressed SecDir review comments by Valery Smyslov.
- Addressed OpsDir review comments by Joe Clarke.
- Addressed GenArt review comments by Vijay Gurbani.

Draft 12 - mbj

- Revised to unambiguously require the use of TLS, while preserving descriptions of precautions needed for non-TLS use in an appendix.

Draft 13 - mbj

- Addressed IESG comments.

Draft 14 - AB

- Revised normative requirements for SET re-transmission to clarify "at least once" delivery expectiations.

- Added non-normative text to Section 4 - Delivery Reliability describing conditions where re-transmission of successfully delivered SETs may occur.

Authors' Addresses

Annabelle Backman (editor)
Amazon

Email: richanna@amazon.com

Michael B. Jones (editor)
Microsoft

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

Marius Scurtescu
Coinbase

Email: marius.scurtescu@coinbase.com

Morteza Ansari
Cisco

Email: morteza.ansari@cisco.com

Anthony Nadalin
Microsoft

Email: tonynad@microsoft.com