Stream:	Internet Engineering Task Force (IETF)				
RFC:	8935				
Category:	Standards Track				
Published:	November 2020				
ISSN:	2070-1721				
Authors:	A. Backman, Ed. *Amazon*	M. Jones, Ed. *Microsoft*	M. Scurtescu *Coinbase*	M. Ansari *Independent*	A. Nadalin *Independent*

RFC 8935
Push-Based Security Event Token (SET) Delivery Using HTTP

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 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/rfc8935.

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 [RFC8936].

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 retransmit 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 retransmit 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** retransmit 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 JSON Web Token (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 Error Codes

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

Error Code	Description
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 Error Codes

Other Error Codes may also be received, as the set of Error Codes is extensible via the IANA "Security Event Token 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 DNS-Based Authentication of Named Entities (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 retransmitting the same SET. Others such as `access_denied` may be transient, for example, if the SET Transmitter refreshes expired credentials prior to retransmission.

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 retransmit 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 retransmitting 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 retransmission 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 Transport Layer Security (TLS) and Datagram Transport Layer Security (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 Error Codes

This document defines Security Event Token Error Codes, for which IANA has created and now maintains a new registry titled "Security Event Token Error Codes". Initial values for the "Security Event Token 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; therefore, they **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 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 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.

Reference
: A reference to the document or documents that define the Security Event Token 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
Reference: Section 2.4 of RFC 8935

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
Reference: Section 2.4 of RFC 8935

Error Code: invalid_issuer
Description: The SET Issuer is invalid for the SET Recipient.
Change Controller: IETF
Reference: Section 2.4 of RFC 8935

Error Code: invalid_audience
Description: The SET Audience does not correspond to the SET Recipient.
Change Controller: IETF
Reference: Section 2.4 of RFC 8935

Error Code: authentication_failed
Description: The SET Recipient could not authenticate the SET Transmitter.
Change Controller: IETF
Reference: Section 2.4 of RFC 8935

Error Code: access_denied
Description: The SET Transmitter is not authorized to transmit the SET to the SET Recipient.
Change Controller: IETF
Reference: Section 2.4 of RFC 8935

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

[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>.

[RFC8936] Backman, A., Ed., Jones, M., Scurtescu, M., Ansari, M., and A. Nadalin, "Poll-Based Security Event Token (SET) Delivery Using HTTP", RFC 8936, DOI 10.17487/RFC8936, October 2020, <https://www.rfc-editor.org/info/rfc8936>.

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 of 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.)

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, Éric Vyncke, and Robert Wilton.

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
Independent
Email: morteza@sharppics.com

Anthony Nadalin
Independent
Email: nadalin@prodigy.net