TLS
Internet-Draft
Intended status: Standards Track
Expires: May 7, 2020

Y. Nir
DellEMC

November 4, 2019

A Flags Extension for TLS 1.3
draft-ietf-tls-tlsflags-01

Abstract

A number of extensions are proposed in the TLS working group that carry no interesting information except the 1-bit indication that a certain optional feature is supported. Such extensions take 4 octets each. This document defines a flags extension that can provide such indications at an average marginal cost of 1 bit each. More precisely, it provides as many flag extensions as needed at 4 + the order of the last set bit divided by 8.

Status of This Memo

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

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

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

This Internet-Draft will expire on May 7, 2020.

Copyright Notice

Table of Contents

1. Introduction

Since the publication of TLS 1.3 ([RFC8446]) there have been several proposals for extensions to this protocol, where the presence of the content-free extension in both the ClientHello and either the ServerHello or EncryptedExtensions indicates nothing except either support for the optional feature or an intent to use the optional feature. Examples:

- An extension that allows the server to tell the client that cross-SNI resumption is allowed: [I-D.sy-tls-resumption-group].

- An extension that is used to negotiate support for authentication using both certificates and external PSKs: [I-D.ietf-tls-tls13-cert-with-extern-psk].

This document proposes a single extension called tls_flags that can enumerate such flag extensions and allowing both client and server to indicate support for optional features in a concise way.

None of the current proposed extensions are such that the server indicates support without the client first indicating support. So as not to preclude future extensions that are so defined, this specification allows the client to send an empty extension, indicating support for TLS flags in general (and presumably some unspecified features in particular). A possible use case for such

extensions is to hide them from passive observers, because the server can send flags in the EncryptedExtensions message, while the client can only send the flags in the clear.

1.1. Requirements and Other Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The term "flag extension" is used to denote an extension where the extension_data field is always zero-length in a particular context, and the presence of the extension denotes either support for some feature or the intent to use that feature.

The term "flag-type feature" denotes an options TLS 1.3 feature the support for which is negotiated using a flag extension, whether that flag extension is its own extension or a value in the extension defined in this document.

2. The tls_flags Extension

This document defines the following extension code point:

```
enum {
   ...
   tls_flags(TBD),
   (65535)
} ExtensionType;
```

This document also defines the data for this extension as a variable-length bit string, allowing for the encoding of an unbounded number of features.

```
struct {
   uint8 flags<0..31>;
} FlagExtensions;
```

The FlagExtensions field 8 flags with each octet, and its length is the minimal length that allows it to encode all of the present flags. Within each octet, the bits are packed such that the first bit is the LSB and the seventh bit is the MSB. The first octet holds flags 0-7, the second octet holds bits 8-15 and so on. For example, if we want to encode only flag number zero, the FlagExtension field will be 1 octet long, that is encoded as follows:

00000001

If we want to encode flags 1 and 5, the field will still be 1 octet long:

00100010

If we want to encode flags 3, 5, and 23, the field will have to be 3 octets long:

00101000 00000000 10000000

Note that this document does not define any particular bits for this string. That is left to the protocol documents such as the ones in the examples from the previous section. Such documents will have to define which bit to set to show support, and the order of the bits within the bit string shall be enumerated in network order: bit zero is the high-order bit of the first octet as the flags field is transmitted.

A client that supports this extension SHALL send this extension with the flags field having bits set only for those extensions that it intends to set. If it does not wish to set any such flags in the ClientHello message, it MAY send the extension empty (with length of zero), or it may omit the extension altogether.

A server that supports this extension and also supports at least one of the flag-type features that use this extension and that were declared by the ClientHello extension SHALL send this extension with the intersection of the flags it supports with the flags declared by the client. The intersection operation MAY be implemented as a bitwise AND. The server may need to send two tls_flags extensions, one in the ServerHello and the other in the EncryptedExtensions message. It is up to the document for the specific feature to

determine whether support should be acknowledged in the ServerHello or the EncryptedExtensions message.

3. IANA Considerations

IANA is requested to assign a new value from the TLS ExtensionType Values registry:

- The Extension Name should be tls_flags
- The TLS 1.3 value should be CH,SH,EE
- The Recommended value should be Y
- The Reference should be this document

IANA is also requested to create a new registry under the TLS namespace with name "TLS Flags" and the following fields:

- Value, which is a number between 0 and 63. All potential values are available for assignment.
- Flag Name, which is a string
- Message, which like the "TLS 1.3" field in the ExtensionType registry contains the abbreviations of the messages that may contain the flag: CH, SH, EE, etc.
- Recommended, which is a Y/N value determined in the document defining the optional feature.
- Reference, which is a link to the document defining this flag.

The policy for this shall be "Specification Required" as described in [RFC8126].

4. Security Considerations

The extension described in this document provides a more concise way to express data that could otherwise be expressed in individual extensions. It does not send in the clear any information that would

otherwise be sent encrypted, nor vice versa. For this reason this extension is neutral as far as security is concerned.

5. Acknowledgements

The idea for writing this was expressed at the mic during the TLS session at IETF 104 by Eric Rescorla.

The current bitwise formatting was suggested on the mailing list by Nikos Mavrogiannopoulos.

Improvement to the encoding were suggested by Ilari Liusvaara, who also asked for a better explanation of the semantics of missing extensions.

6. References

6.1. 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>.

[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>.

[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>.

6.2. Informative References

[I-D.ietf-tls-tls13-cert-with-extern-psk] Housley, R., "TLS 1.3 Extension for Certificate-based Authentication with an External Pre-Shared Key", draft-ietf-tls-tls13-cert-with-extern-psk-02 (work in progress), May 2019.

[I-D.sy-tls-resumption-group]
Sy, E., "TLS Resumption across Server Name Indications for TLS 1.3", draft-sy-tls-resumption-group-00 (work in progress), March 2019.

[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>.

Appendix A. Change Log

RFC EDITOR: PLEASE REMOVE THIS SECTION AS IT IS ONLY MEANT TO AID THE WORKING GROUP IN TRACKING CHANGES TO THIS DOCUMENT.

draft-ietf-tls-tlsflags-01 allows server-only flags and allows the client to send an empty extension. Also modified the packing order of the bits.

draft-ietf-tls-tlsflags-00 had the same text as draft-nir-tls-tlsflags-02, and was re-submitted as a working group document following the adoption call.

Version -02 replaced the fixed 64-bit string with an unlimited bitstring, where only the necessary octets are encoded.

Version -01 replaced the enumeration of 8-bit values with a 64-bit bitstring.

Version -00 was a quickly-thrown-together draft with the list of supported features encoded as an array of 8-bit values.

Author's Address

Yoav Nir
DellEMC
9 Andrei Sakharov St
Haifa 3190500
Israel

Email: ynir.ietf@gmail.com