TLS Working Group T. Fossati
Internet-Draft Nokia
Updates: 5246, 6347 (if approved) N. Mavrogiannopoulos
Intended status: Standards Track RedHat
Expires: September 4, 2018 March 03, 2018

Record Header Extensions for DTLS
draft-fossati-tls-ext-header-01

Abstract

This document proposes a mechanism to extend the record header in DTLS. To that aim, the DTLS header is modified as follows: the length field is trimmed to 15 bits, and the length's top bit is given the "record header extension indicator" semantics, allowing a sender to signal that one or more record header extensions have been added to this record. We define the generic format of a record header extension and the general rules associated with its handling. Any details regarding syntax, semantics and negotiation of a specific record header extension, are left to future documents.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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 [RFC2119].

2. Length Redefined

DTLS ([RFC6347], [I-D.ietf-tls-dtls13]) requires the size of record payloads to not exceed 2^14 bytes - plus a small amount that accounts for compression or AEAD expansion. This means that the first bit in the length field of the DTLS record header is, in fact, unused.

The proposal (Figure 1) is to shorten the length field to 15 bits and use the top bit (E) to signify the presence / absence of a record header extension.

```
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+
   |  ContentType  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |        ProtocolVersion        |             epoch             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         sequence_number                       |
   +                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                               |E|           length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~             (zero or more) Extension Header(s)                ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         Payload (including optional MAC and padding)          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Length redefined

Length counts the bytes of Payload and of all record header extensions that are added to this record (possibly none).

In the reminder, the top bit is called the E-bit.

3. Record Header Extension

3.1. Format

If the E-bit is asserted, then a record header extension is appended to the regular header with the following format:

```
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+------------+
   |M| Type  |       Length        | Value ... |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+------------+
```

Where:

- o M(ore) has the same semantics as the E-bit in the regular header - i.e.: if it is asserted then another extension header follows this one;
- o Type is a fixed length (4-bits) field that defines the way Value has to be interpreted;
- o Length is the size of Value in bytes. It uses 11 bits, therefore allowing a theoretical maximum size of 2047 bytes for any record header extension;
- o Value is the record header extension itself.

The fact that Type only allows 16 record header extension is a precise design choice: the allocation pool size is severely constrained so to raise the entry bar for any new record header extension.

3.2. Negotiation

A record header extension is allowed only if it has been negotiated via a companion DTLS extension.

An endpoint MUST NOT send a record header extension that hasn't been successfully negotiated with the receiver.

An endpoint that receives an unexpected record header extension MUST abort the session.

Record header extensions MUST NOT be sent during the initial handshake phase.

3.3. Backwards Compatibility

A legacy endpoint that receives a record header extension will interpret it as an invalid length field ([RFC6347], [I-D.ietf-tls-dtls13]) and abort the session accordingly.

Note that this is equivalent to the behaviour of an endpoint implementing this spec which receives a non-negotiated record header extension.

3.4. Use with Connection ID

A plausible use of this mechanism is with the CID extension defined in [I-D.ietf-tls-dtls-connection-id].

In that case, the companion record header extension could be defined as follows:

- Type: 0x0 (i.e., CID record header extension);

- Value: the CID itself

A DTLS 1.2 record carrying a CID "AB" would be formatted as in Figure 2:

- E=1

- Type=0x0

o Length=0x002

o Value=0x4142

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+
|  ContentType  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Version            |             Epoch             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |1|           Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|  0x0  |       0x002         |            0x4142             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Payload (including optional MAC and padding)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: CID header example

Note that, compared to all other possible ways to express presence/absence of a CID field within the constraints of the current header format (e.g., bumping the Version field, assigning new ContentType(s), using an invalid length), an ad hoc record header extension provides a cleaner approach that can be used with any TLS version at a reasonable cost - an overhead of 2 bytes per record.

4. Security Considerations

An on-path active attacker could try and modify an existing record header extension, insert a new record header extension in an existing session, or alter the result of the negotiation in order to add or remove arbitrary record header extensions. Given the security properties of DTLS, none of the above can be tried without being fatally noticed by the endpoints.

A passive on-path attacker could potentially extrapolate useful knowledge about endpoints from the information encoded in a record header extension (see also Section 5).

5. Privacy Considerations

The extent and consequences of metadata leakage from endpoints to path when using a certain record header extension SHALL be assessed in the document that introduces this new record header extension. If needed, the document SHALL describe the relevant risk mitigations.

6. IANA Considerations

This document defines a new IANA registry that, for each new record header extension, shall provide its Type code-point.

7. Acknowledgements

Thanks to Adam Langley and Yoav Nir for comments and discussions that have helped shaping this document.

This work is partially supported by the European Commission under Horizon 2020 grant agreement no. 688421 Measurement and Architecture for a Middleboxed Internet (MAMI). This support does not imply endorsement.

8. References

8.1. Normative References

[I-D.ietf-tls-dtls13]
Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", draft-ietf-tls-dtls13-22 (work in progress), November 2017.

[I-D.ietf-tls-tls13]
Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", draft-ietf-tls-tls13-25 (work in progress), March 2018.

[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>.

[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>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

8.2. Informative References

[I-D.ietf-tls-dtls-connection-id]
Rescorla, E., Tschofenig, H., Fossati, T., and T. Gondrom, "The Datagram Transport Layer Security (DTLS) Connection Identifier", draft-ietf-tls-dtls-connection-id-00 (work in progress), December 2017.

Authors' Addresses

Thomas Fossati
Nokia

Email: thomas.fossati@nokia.com

Nikos Mavrogiannopoulos
RedHat

Email: nmav@redhat.com