Internet Engineering Task Force (IETF)
Request for Comments: 7685
Updates: 5246
Category: Standards Track
ISSN: 2070-1721

A. Langley
Google Inc

October 2015

A Transport Layer Security (TLS) ClientHello Padding Extension

Abstract

This memo describes a Transport Layer Security (TLS) extension that can be used to pad ClientHello messages to a desired size.

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 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7685.

Table of Contents

1. Introduction

Successive TLS [RFC5246] versions have added support for more cipher suites and, over time, more TLS extensions have been defined. This has caused the size of the TLS ClientHello to grow, and the additional size has caused some implementation bugs to come to light. At least one of these implementation bugs can be ameliorated by making the ClientHello even larger. This is desirable given that fully comprehensive patching of affected implementations is difficult to achieve.

This memo describes a TLS extension that can be used to pad a ClientHello to a desired size in order to avoid implementation bugs caused by certain ClientHello sizes.

2. Requirements Notation

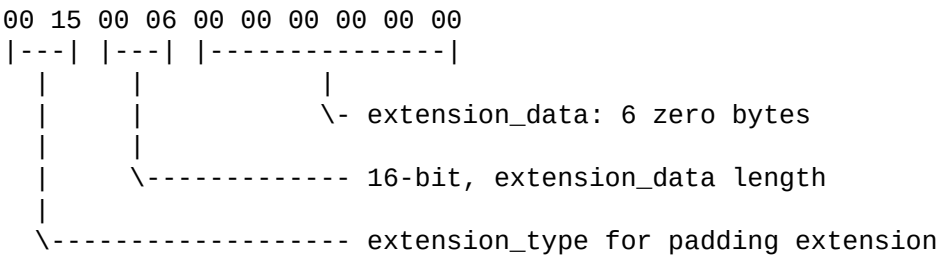
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Padding Extension

A new extension type ("padding(21)") is defined and MAY be included by the client in its ClientHello message.

```
enum {
    padding(21), (65535)
} ExtensionType;
```

The "extension_data" for the extension consists of an arbitrary number of zero bytes. For example, the smallest "padding" extension is four bytes long and is encoded as 0x00 0x15 0x00 0x00. A ten-byte extension would include six bytes of "extension_data" and would be encoded as:



The client MUST fill the padding extension completely with zero bytes, although the padding extension_data field may be empty.

The server MUST NOT echo the extension.

4. Example Usage

As an example, consider a client that wishes to avoid sending a ClientHello with a TLSCiphertext.length between 256 and 511 bytes (inclusive). This case is considered because at least one TLS implementation is known to hang the connection when such a ClientHello record is received.

After building a ClientHello as normal, the client can add four bytes to the length (to account for the "msg_type" and "length" fields of the handshake protocol) and test whether the resulting length falls into that range. If it does, a padding extension can be added in order to push the length to (at least) 512 bytes.

Note that if the original ClientHello size was between 505 and 507 bytes, then, with the handshake protocol overhead, the record payload would be between 509 and 511 bytes long. Since it’s not possible for an extension to take less than four bytes of space, the additional padding would have to expand the ClientHello record payload beyond 512 bytes in these cases.

5. Security Considerations

The contents of the padding extension could be used as a covert channel. In order to prevent this, the contents are required to be all zeros, although the length of the extension can still be used as a much smaller covert channel.

6. IANA Considerations

IANA has permanently registered value 21 (padding) in the "ExtensionType Values" registry.

7. 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, <http://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, <http://www.rfc-editor.org/info/rfc5246>.

Acknowledgements

The author gratefully acknowledges the contributions of Wan-Teh Chang and the suggestions of Eric Rescorla.

Author's Address

Adam Langley
Google Inc

Email: agl@google.com