Internet Engineering Task Force (IETF)
Request for Comments: 8740
Updates: 7540
Category: Standards Track
ISSN: 2070-1721

D. Benjamin
Google LLC
February 2020

Using TLS 1.3 with HTTP/2

Abstract

This document updates RFC 7540 by forbidding TLS 1.3 post-handshake authentication, as an analog to the existing TLS 1.2 renegotiation restriction.

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[1].

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/rfc8740[2].

Copyright Notice

[1] https://www.rfc-editor.org/rfc/rfc7841.html#section-2
[2] https://www.rfc-editor.org/info/rfc8740
[3] https://trustee.ietf.org/license-info

Table of Contents

1. Introduction

TLS 1.2 [RFC5246] and earlier versions of TLS support renegotiation, a mechanism for changing parameters and keys partway through a connection. This was sometimes used to implement reactive client authentication in HTTP/1.1 [RFC7230], where the server decides whether or not to request a client certificate based on the HTTP request.

HTTP/2 [RFC7540] multiplexes multiple HTTP requests over a single connection, which is incompatible with the mechanism above. Clients cannot correlate the certificate request with the HTTP request that triggered it. Thus, Section 9.2.1 of [RFC7540] forbids renegotiation.

TLS 1.3 [RFC8446] removes renegotiation and replaces it with separate post-handshake authentication and key update mechanisms. Post-handshake authentication has the same problems with multiplexed protocols as TLS 1.2 renegotiation, but the prohibition in [RFC7540] only applies to renegotiation.

This document updates HTTP/2 [RFC7540] to similarly forbid TLS 1.3 post-handshake authentication.

2. Requirements Language

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.

3. Post-Handshake Authentication in HTTP/2

HTTP/2 servers MUST NOT send post-handshake TLS 1.3 CertificateRequest messages. HTTP/2 clients MUST treat such messages as connection errors (see Section 5.4.1 of [RFC7540]) of type PROTOCOL_ERROR.

[RFC7540] permitted renegotiation before the HTTP/2 connection preface to provide confidentiality of the client certificate. TLS 1.3 encrypts the client certificate in the initial handshake, so this is no longer necessary. HTTP/2 servers MUST NOT send post-handshake TLS 1.3 CertificateRequest messages before the connection preface.

The above applies even if the client offered the `post_handshake_auth` TLS extension. This extension is advertised independently of the selected Application-Layer Protocol Negotiation (ALPN) protocol [RFC7301], so it is not sufficient to resolve the conflict with HTTP/2. HTTP/2 clients that also offer other ALPN protocols, notably HTTP/1.1, in a TLS ClientHello MAY include the `post_handshake_auth` extension to support those other protocols. This does not indicate support in HTTP/2.

4. Other Post-Handshake TLS Messages in HTTP/2

[RFC8446] defines two other messages that are exchanged after the handshake is complete: KeyUpdate and NewSessionTicket.

KeyUpdate messages only affect TLS itself and do not require any interaction with the application protocol. HTTP/2 implementations MUST support key updates when TLS 1.3 is negotiated.

NewSessionTicket messages are also permitted. Though these interact with HTTP when early data is enabled, these interactions are defined in [RFC8470] and are allowed for in the design of HTTP/2.

Unless the use of a new type of TLS message depends on an interaction with the application-layer protocol, that TLS message can be sent after the handshake completes.

5. Security Considerations

This document resolves a compatibility concern between HTTP/2 and TLS 1.3 when supporting post-handshake authentication with HTTP/1.1. This lowers the barrier for deploying TLS 1.3, a major security improvement over TLS 1.2.

6. IANA Considerations

This document has no IANA actions.

7. References

7.1. 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/rfc>.

[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/rfc>.

[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/rfc>.

[RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301, July 2014, <https://www.rfc-editor.org/info/rfc>.

[RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <https://www.rfc-editor.org/info/rfc>.

[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/rfc>.

[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/rfc>.

7.2. Informative References

[RFC8470] Thomson, M., Nottingham, M., and W. Tarreau, "Using Early Data in HTTP", RFC 8470, DOI 10.17487/RFC8470, September 2018, <https://www.rfc-editor.org/info/rfc>.

Author's Address

David Benjamin
Google LLC
EMail: davidben@google.com