Workgroup: Network Working Group
Internet-Draft:
draft-thomson-httpbis-h2-0rtt-00
Updates: 7540, 8441 (if approved)
Published: 16 December 2020
Intended Status: Standards Track
Expires: 19 June 2021
Authors: M. Thomson
Mozilla

Optimizations for Using TLS Early Data in HTTP/2

Abstract

This proposes an extension to HTTP/2 that enables the use of server settings by clients that send requests in TLS early data. In particular, this allows extensions to the protocol to be used.

This amends the definition of settings defined in RFC 7540 and RFC 8441 and introduces new registration requirements for HTTP/2 settings.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the HTTP Working Group mailing list (ietf-http-wg@w3.org), which is archived at https://lists.w3.org/Archives/Public/ietf-http-wg/.

Source for this draft and an issue tracker can be found at https://github.com/martinthomson/h2-0rtt.

Status of This Memo

Table of Contents

1. Introduction
2. Conventions and Definitions
3. EARLY_DATA_SETTINGS Setting
   3.1. Server Handling
   3.2. Client Handling
   3.3. Use for Resumption
4. Settings in Early Data
   4.1. Existing HTTP/2 Settings
   4.2. CONNECT Protocol
   4.3. Replay Attack Risk
   4.4. Advertising Less-Permissive Values
5. Security Considerations
6. IANA Considerations
7. References
   7.1. Normative References
   7.2. Informative References
Author's Address

1. Introduction

HTTP/2 [HTTP2] does not include any special provisions for the use of TLS early data as it was published prior to the introduction the feature in TLS 1.3 [TLS]. As a result, when using HTTP/2 with TLS early data, clients are forced to assume defaults for the server configuration.

Using the initial value of settings can adversely affect performance as it can take an additional round trip or two to receive the connection preface from the server. This is especially noticeable for new features that are added using extensions. Clients that wish to use extensions therefore have to deal with extended delays before they can confirm server support for the extension.

In contrast, HTTP/3 [HTTP3] was defined for use with QUIC [QUIC], which includes early data (or 0-RTT) as a core features. The use in HTTP/3 demonstrates the value of access to non-default values of server configuration, especially for performance.

This document defines a new setting for servers and clients to indicate a willingness to remember settings from a previous connection when attempting TLS early data. This allows clients to rely on capabilities established in a previous connection. This also offers servers the ability to place tighter restrictions on use of early data than the initial values of settings otherwise allows.

2. Conventions and Definitions

This document relies on concepts from [HTTP2] and [TLS].

3. EARLY_DATA_SETTINGS Setting

The EARLY_DATA_SETTINGS setting (0xTBD) is sent to indicate support for remembering the value of settings in TLS early data.

A server that advertises a value for EARLY_DATA_SETTINGS of 1 MUST remember all settings defined as being applicable to early data; see Section 4. A client that advertises a value for EARLY_DATA_SETTINGS of 1 and has received a value of 1 from a server MUST respect these settings when attempting early data.

3.1. Server Handling

An EARLY_DATA_SETTINGS value of 1 indicates that the server will respect any settings that can apply to early data if it accepts the early data; see Section 4. A value of 0, the initial value, indicates that settings assume their initial values for resumed connections (that is, the default behavior in HTTP/2).

Any session tickets that are sent by the server subsequent to a SETTINGS frame containing EARLY_DATA_SETTINGS set to 1 are affected by this feature. The value of all applicable settings apply to each session ticket as TLS NewSessionTicket messages are received.

In addition, setting a value of 1 in the SETTINGS frame that is part of the connection preface has the effect of applying to all session tickets sent prior to that point; the settings that are used for those session tickets is taken from the connection preface.

Initial values for settings are used if those settings are not explicitly sent in a SETTINGS frame.

A server does not need to wait for a SETTINGS acknowledgment before it sends a TLS NewSessionTicket message. Values from SETTINGS frames apply immediately to any subsequent TLS NewSessionTicket messages.

Note: As the arrival of SETTINGS frames is strictly ordered with respect to TLS NewSessionTicket messages, this ensures that the value of settings that apply to each session ticket is unambiguous.

Once set to a value of 1, a server can set this value to 0 in subsequent SETTINGS frames to indicate that updated settings values do not apply to early data. This could be used by a server to set values that are more permissive than it might be willing to accept for early data.

A server that might have set EARLY_DATA_SETTINGS to 1 and does not remember the value of settings MUST reject early data. Similarly, a server that cannot respect the values that it previously set MUST reject early data.

A server that advertises a value of 1 MUST remember settings even if the client does not indicate support for EARLY_DATA_SETTINGS.

3.2. Client Handling

A client advertises a value of 1 for EARLY_DATA_SETTINGS to indicate that it will respect the settings that a server sets when attempting to use early data if the server also advertises a value of 1; see Section 4.4.

A client that advertises a value of 1 for EARLY_DATA_SETTINGS MUST remember the value of all applicable server settings at the time that a TLS NewSessionTicket was received if the server settings include a a value of 1 for EARLY_DATA_SETTINGS. These settings values are then used for server settings in place of initial values if early data is accepted by the server.

A client MUST NOT set a value of 0 for EARLY_DATA_SETTINGS after it advertises a value of 1. A server can treat a change in the value of EARLY_DATA_SETTINGS from 1 to 0 as a connection error (see Section 5.4.1 of [HTTP2]) of type PROTOCOL_ERROR.

3.3. Use for Resumption

It might have been possible to define a similar setting to indicate that a server would respect settings for TLS session resumption more generally. This would have the benefit of providing starting values for clients that differ from the protocol-defined initial values.

However, resumption does not come with a clear rejection signal in the same way as early data. Servers would not have any way to invalidate previous settings short of rejecting resumption, which could have undesirable performance consequences. Furthermore, a setting of that type would be difficult for clients to adapt to as many clients do not currently condition their behavior on whether the underlying TLS connection is resumed or full.

There are potential advantages from the mechanism in this draft as it provides a way for clients to use non-initial values for settings even where 0.5-RTT data is not sent by the server. Clients that want the performance gains provided by the EARLY_DATA_SETTINGS setting, but do not want any exposure to replay attack can use early data and limit their use of that to sending the connection preface, which carries no risk from replay.

4. Settings in Early Data

Some settings cannot apply during TLS early data. Other settings might represent too much of a risk of replay attack. For a setting to be usable in early data, a definition MUST be provided for how the value is handled. This definition MUST include either an analysis showing that use of the setting in early data is safe, or rules for managing the risk of replay attack arising from its use; see Section 4.3 for details.

Exposure to replay attacks does not automatically disqualify settings from use with EARLY_DATA_SETTINGS. As noted in Section 3.3, there is value in being able to use remembered values of settings in place of initial values, even if the functions enabled by the setting cannot be used in early data.

Table 1 in Section 6 contains a summary of existing settings and whether they are remembered when EARLY_DATA_SETTINGS is enabled.

4.1. Existing HTTP/2 Settings

This document amends the definition of extensions defined in [HTTP2] to permit their use with early data.

The ENABLE_PUSH setting only applies to clients. Though [HTTP2] does not prohibit servers from advertising a value, there is no value in doing so. ENABLE_PUSH is marked as not remembered for early data.

The other settings defined in [HTTP2] all represent resource limits that could apply to early data. These values can all be remembered and applied to early data. As resource limits, their use does not carry actionable information and so none of these settings cannot contribute to the risk of replay attacks.

A server that advertises a value for EARLY_DATA_SETTINGS of 1 MUST remember all settings defined in [HTTP2], aside from ENABLE_PUSH. A client that advertises a value for EARLY_DATA_SETTINGS of 1 and receives a value of 1 MUST respect these settings when attempting early data.

4.2. CONNECT Protocol

The setting defined in [HTTP-WS] governs CONNECT requests. This document establishes this setting as applicable to early data.

Using CONNECT to establish a TCP connection is observable behavior that might in itself comprise a risk of replay as it would allow an attacker to use replay attacks to learn about any CONNECT requests were included in early data. A server could tentatively allocate a connection that was pre-emptively made to a CONNECT request that arrives in early data without significant risk of leaking significant information, but establishing a connection in reaction to the request would leak information.

In addition, any actions that are taken based on any early data sent in the CONNECT tunnel presents a potential risk in the event of a replay attack. Even connection establishment might result in side-effects that can be exploited in the event of a replay attack.

For this reason, a client that sends a CONNECT request in early data cannot expect the request to be processed until the handshake is complete. A server MUST delay processing of any CONNECT request until the handshake is complete, or reject any attempt with a 425 (Too Early) status code.

Though this limits the applicability of the capability, SETTINGS_ENABLE_CONNECT_PROTOCOL is marked as requiring servers to remember the value when accepting early data. This allows clients to send requests in early data, or before receiving the connection preface from the server.

4.3. Replay Attack Risk

Use of TLS early data requires careful consideration of the potential for replay attack. [HTTP-REPLAY] provides a discussion about what this means for HTTP requests. That advice applies to settings that might affect the generation or handling of HTTP requests.

Extensions to HTTP/2 that are used by a client before the handshake completes might not be limited to those that affect requests.

Extensions that are limited in effect to the state of the HTTP/2 connection have limited exposure to replay attacks. Replayed

connection attempts cannot be completed successfully, so any effect is discarded.

Extensions that might affect requests or result in other activity not limited to connection state MUST define rules for how the risk of replay attack is managed. Techniques similar to those in [HTTP-REPLAY], such as deferral of processing and rejection could be used. Extensions that do not include describe any analysis of or mitigations for the risk of replay attack MUST indicate in their definition that they cannot be used in 0-RTT.

4.4. Advertising Less-Permissive Values

One potential value of advertising the EARLY_DATA_SETTINGS setting is that a server is able to restrict the resources that a client can consume with early data. Though TLS provides the max_early_data_size field in the early_data extension, which limits the total data that the server can accept, there might be other resources that a server does not wish to commit.

If a client does not support EARLY_DATA_SETTINGS, it could consume resources up to the limits implied by initial values of settings. This includes a number of request streams that is only bounded by the value of max_early_data_size.

A server might choose to condition support for early data on client support for EARLY_DATA_SETTINGS, only sending session tickets that permit use of early data after receiving a value of 1. In this way, a server can rely on clients respecting any stricter limits to resource usage that are advertised.

A server cannot rely on being able to limit resource usage in this way beyond early data. The server might be forced to reject early data, at which point the client uses the initial values for settings.

5. Security Considerations

The potential for replay attacks on early data is significant and needs consideration; see Section 4.3 for details.

An endpoint that offers this setting requires a larger amount of state associated with sessions that might be resumed with early data. This state is bounded in size and can be offloaded using session tickets, so this is expected to be manageable.

6. IANA Considerations

The "HTTP/2 Settings" registry established in HTTP/2 [HTTP2] is modified to include a new field for each entry, titled "Early Data". This field has one of two values:

* A value of "Y" indicates that the value of this setting advertised by a server is remembered by that if it advertises the EARLY_DATA_SETTINGS setting. In so doing, clients can rely on the value of the setting when attempting to use TLS early data. Clients MUST remember settings values and respect any values it has remembered when attempting to use early data.

* A value of "N" indicates that the setting does not need to be remembered by a server or respected by a client when accepting or attempting early data. The client needs to observe initial values for settings until the server sends its first SETTINGS frame.

New registrations to this registry MUST specify a value for this field.

Initial values for existing values are listed in Table 1.

Code	Name	Early Data
0x1	HEADER_TABLE_SIZE	Y
0x2	ENABLE_PUSH	N
0x3	MAX_CONCURRENT_STREAMS	Y
0x4	INITIAL_WINDOW_SIZE	Y
0x5	MAX_FRAME_SIZE	Y
0x6	MAX_HEADER_LIST_SIZE	Y
0x8	SETTINGS_ENABLE_CONNECT_PROTOCOL	Y
0x10	TLS_RENEG_PERMITTED	N

Table 1: Early Data Values for Settings

7. References

7.1. Normative References

[HTTP-WS] McManus, P., "Bootstrapping WebSockets with HTTP/2", RFC 8441, DOI 10.17487/RFC8441, September 2018, <https://www.rfc-editor.org/info/rfc8441>.

[HTTP2] 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/rfc7540>.

[QUIC] Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", Work in Progress, Internet-Draft, draft-ietf-quic-transport-33, 13 December 2020, <http://www.ietf.org/internet-drafts/draft-ietf-quic-transport-33.txt>.

[TLS] 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>.

7.2. Informative References

[HTTP-REPLAY] 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/rfc8470>.

[HTTP3] Bishop, M., "Hypertext Transfer Protocol Version 3 (HTTP/3)", Work in Progress, Internet-Draft, draft-ietf-quic-http-32, 20 October 2020, <http://www.ietf.org/internet-drafts/draft-ietf-quic-http-32.txt>.

Author's Address

Martin Thomson
Mozilla

Email: mt@lowentropy.net