tls
Internet-Draft
Intended status: Experimental
Expires: September 2, 2019

E. Sy
University of Hamburg

March 01, 2019

TLS Resumption across Server Name Indications for TLS 1.3
draft-sy-tls-resumption-group-00

Abstract

This document defines a mechanism for resuming a TLS 1.3 session across different Server Name Indications.

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 September 2, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

Most web transactions are short transfers that are significantly delayed by the TLS connection establishment. To accelerate the connection establishment, TLS 1.3 [RFC8446] and its predecessors provide session resumption mechanisms. They abbreviate the TLS handshake based on a shared secret exchanged during a prior TLS session between client and server. In total, these resumption handshakes significantly reduce computational overhead for cryptographic operations and save up to one round-trip compared to the full TLS connection establishment.

TLS 1.3 [RFC8446] allows resumption handshakes across Server Name Indications (SNIs) when they share the same TLS certificate. However, TLS 1.3 recommends not to use TLS resumptions across SNIs to avoid loosing a single-use ticket in case of a failed resumption attempt. This practice requires costly full TLS connection establishments in situations where a performance-optimized resumption handshake across SNI values would be possible. To illustrate this performance limitation, we describe the common situation of a redirected web request. We assume that the hostname example.com redirects to www.example.com and both hostnames are operated by the same entity and use the same certificate for their authentication. A client requesting www.example.com via this redirect requires two full TLS handshakes following the recommendation of TLS 1.3 [RFC8446]. Using resumption across SNI values, the later full handshake can be converted to a performance-optimized resumed handshake. A comprehensive study of the performance benefits of resumptions across SNI values for popular websites can be found in [PERF].

This document defines a mechanism to inform the client in between which SNI values TLS resumptions are supported. This information enables the client to use resumption across SNI values only in situations where the chance of a successful resumption handshake is high.

2. Conventions and Definitions

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. Overview on Resumptions across SNI values

When a client wants to form a TLS connection to a server, it indicates support for the "resumption_group" extension in the ClientHello message. To signal its support for this extension type, the server returns the "resumption_group" extension with an empty data field.

The client is now aware, that all SNI values for which the presented server certificate is valid, form a TLS resumption group. Thus, resumption tickets issued by a group member are designated to be used to establish resumed connections to any member of the same group.

4. The "resumption_group" Extension

This extension carries no data as defined in the following ResumptionGroup structure:

```
struct {
} ResumptionGroup;
```

4.1. Client Behavior

To indicate support for the "resumption_group" extension, the client sends this extension type within the initial ClientHello message to the server.

Upon receiving the server's response, the client checks whether the "resumption_group" extension is present in the extension list of the server's CertificateEntry (see Section 4.2.2 of [RFC6066]).

If this extension type is not included in the response of the server, then the client reasons that the server is not configured to support

the "resumption_group" extension and proceeds with a normal handshake.

Otherwise, the client proceeds with a normal connection establishment and associates all retrieved resumption tickets to the corresponding resumption group. This resumption group is formed of all SNI values that are valid for the presented server certificate.

To establish a resumed connection to any SNI value included in a resumption group, the client uses a resumption ticket associated to the same group. The Client Hello of a resumed handshake MUST NOT include the "resumption_group" extension.

Tickets received during a resumed connection MUST be associated to the same resumption group of the ticket that was used during the establishment of this connection.

If a SNI value is a member of multiple resumption groups, then the client is recommended to use the freshest valid ticket for a resumption handshake. It is assumed, that fresher resumption tickets are more likely to be accepted by the respective server.

According to [RFC8446], clients MUST NOT cache tickets longer than seven days.

Note, that TLS resumption enables a server to link resumed connections to the same client. A study on the feasibility of this tracking mechanism can be found in [TRAC]. To protect the client's privacy against tracking via this mechanism, it is RECOMMENDED to cache resumption tickets only for ten minutes.

4.2. Server Behavior

Upon receiving an initial Client Hello message, the server validates if the client provided an extension of the type "resumption_group".

If the "resumption_group" extension is not listed by the client, then the server's response MUST NOT include an entry for this extension type. Otherwise, the server includes the "resumption_group" extension in the extension list of the server's CertificateEntry, to signal support for resumptions across SNI values. Subsequently, the server proceeds with a normal handshake.

This extension type does not affect the server behavior for resumed connection establishments.

5. Expectations on Certificates

This "resumption_group" extension forms the resumption group based on the SNI values that are valid for the server's certificate. To optimize the performance benefit of this extension, the server's certificate is RECOMMENDED to only include SNI values that mutually support the resumption of their TLS connections. Otherwise, the client's resumption attempt across SNI values will fail if the server does not support this practice. Note, that each failed resumption handshake uses up a single-use resumption ticket. As a result, these failed attempts might use up all cached single-use tickets, which hinders the client to establish performance-optimized resumption handshakes to legitimate SNI values.

6. Compatibility Issues with Middleboxes

[RFC8446]; Section 9.3 requires MITM proxies to remove any extensions they do not understand. If a conformant MITM proxy does not support this extension, it will remove this extension type from the Client Hello. As a result, the server reacts as if it is not supporting this extension type.

7. Security Considerations

Clients MUST only resume to a new SNI value if this SNI value is valid for the server certificate presented in the original connection. To facilitate a correct implementation of this requirement, the resumption group is identical to the list of SNI values valid for a specific server certificate. Note, that the security of TLS resumptions across different SNI values is also discussed in Section 4.6.1 of [RFC8446].

8. IANA Considerations

TODO IANA needs to be requested to create an entry, resumption_group, in the existing registry for ExtensionType (defined in [RFC8446]), with "TLS 1.3" column values being set to "CH, EE", and "Recommended" column being set to "Yes".

9. References

9.1. 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>.

9.2. Informative References

[PERF] Sy, E., Moennich, M., Mueller, T., Federrath, H., and M. Fischer, "Enhanced Performance for the encrypted Web through TLS Resumption across Hostnames", 2019, <https://arxiv.org/pdf/1902.02531.pdf>.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <https://www.rfc-editor.org/info/rfc6066>.

[TRAC] Sy, E., Burkert, C., Federrath, H., and M. Fischer, "Tracking Users across the Web via TLS Session Resumption", 2018, <https://arxiv.org/pdf/1810.07304.pdf>.

Acknowledgments

Tobias Mueller and Christian Burkert provided ideas for this document.

Author's Address

Erik Sy
University of Hamburg

Email: tls@erik-sy.de