Workgroup: TLS Working Group
Internet-Draft:
draft-ietf-tls-cross-sni-resumption-02
Published: 6 December 2021
Intended Status: Standards Track
Expires: 9 June 2022
Authors: V. Vasiliev
Google

# Transport Layer Security (TLS) Resumption across Server Names

## Abstract

This document specifies a way for the parties in the Transport Layer Security (TLS) protocol to indicate that an individual session ticket can be used to perform resumption even if the Server Name of the new connection does not match the Server Name of the original.

## Discussion Venues

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

Discussion of this document takes place on the TLS Working Group mailing list (tls@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/tls/.

Source for this draft and an issue tracker can be found at https://github.com/vasilvv/tls-cross-sni-resumption.

## Status of This Memo



## Copyright Notice

**Table of Contents**



## 1. Introduction

Transport Layer Security protocol [RFC8446] allows the clients to use an abbreviated handshake in cases where the client has previously established a secure session with the same server. This mechanism is known as "session resumption", and its positive impact on performance makes it desirable to be able to use it as frequently as possible.

Modern application-level protocols, HTTP in particular, often require accessing multiple servers within a single workflow. Since the identity of the server is established through its certificate, in the ideal case, the resumption would be possible to all of the domains for which the certificate is valid (see [PERF] for a survey of potential practical impact of such approach). TLS, starting with version 1.3, defines the SNI value to be a property of an individual connection that is not retained across sessions ([RFC8446], Section 4.2.11). However, in the absence of additional signals, it discourages using a session ticket when the SNI value does not match ([RFC8446], Section 4.6.1), as there is normally no reason to assume that all servers sharing the same certificate would also share the same session keys. The extension defined in this document allows the server to provide such a signal in-band.

## 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. The Flag

Resumption across server names is negotiated using the TLS flags extension [I-D.draft-ietf-tls-tlsflags]. The server MAY send a resumption_across_names(8) flag in a NewSessionTicket message; the flag is an assertion by the server that any server for any identity presented in its certificate would be capable of accepting that ticket. A client receiving a ticket with this flag MAY attempt resumption for any server name corresponding to an identity in the server certificate even if the new server name value does not match the one used in the original session; note that this requires the client to retain the list of the names specified in the original server certificate. The flag cannot be used in TLS versions before 1.3, as the NewSessionTicket message does not exist in those versions.

## 4. Security Considerations

This document does not alter any of the security requirements of [RFC8446], but merely lifts a performance-motivated "SHOULD NOT" recommendation from Section 4.6.1. Notably, it still relies on the client ensuring that the server certificate is valid for the new SNI at the time of session resumption.

If the original server's assertion regarding supporting cross-name resumption turns out to be incorrect, a different server that receives a misdirected ticket will not be able to decrypt it and will therefore be unable to resume. The protocol will gracefully recover from such situations, as session resumption may be safely rejected for any reason. However, such misconfiguration will waste tickets stored in the client's cache, as TLS tickets may be single-use, leading to a potential performance regression.

When providing the SNI value to the application, TLS 1.3 requires the value from the most recent ClientHello to be used ([RFC8446], Section 4.6.1). If the server TLS implementation violates that requirement and instead reports the SNI value of the original session, this can lead to a confusion attack where the client and the server disagree on the server name being used (similar to the attacks described in [DB15]). The implementers MUST ensure that this

aspect of SNI processing is handled correctly before enabling cross-name resumption.

Cross-domain resumption implies that any certificate the client provides for one host would become available to the other hosts using the same server certificate. Because of that, when performing cross-domain resumption, the client MUST use the same policy on whether to present said certificate to the server as if it were a new TLS session. For instance, if the client would show a certificate choice prompt for every individual domain it connects to, it MUST show that prompt for the new host when performing cross-domain resumption.

Cross-domain resumption, like other similar mechanisms (e.g. cross-domain HTTP connection reuse), can incentivize the server deployments to create server certificates valid for a wider range of domains than they would otherwise. However, any increase in the scope of a certificate comes at a cost: the wider is the scope of the certificate, the wider is the impact of the key compromise for that certificate. In addition, creating a certificate that is valid for multiple hostnames can lead to complications if some of those hostnames change ownership, or otherwise require a different operational domain.

Session tickets can contain arbitrary information, and thus could be potentially used to re-identify a user from a previous connection. Cross-domain resumption expands the potential list of servers to which an individual ticket could be presented. Client applications should partition the session cache between connections that are meant to be uncorrelated. For example, the Web use case uses network partition keys to separate cache lookups [FETCH].

## 5. IANA Considerations

IANA (will add/has added) the following entry to the "TLS Flags" table of the "Transport Layer Security (TLS) Extensions" registry:

**Value** 0x8

**Flag Name** resumption_across_names

**Message** NST

**Recommended** N

**Reference** This document

## 6. References

### 6.1. Normative References

**[I-D.draft-ietf-tls-tlsflags]**
Nir, Y., "A Flags Extension for TLS 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-tlsflags-07, 25 October 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-tlsflags-07>.

**[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/rfc/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/rfc/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/rfc/rfc8446>.

### 6.2. Informative References

**[DB15]** Delignat-Lavaud, A. and K. Bhargavan, "Network-based Origin Confusion Attacks against HTTPS Virtual Hosting", 15 March 2015.

**[FETCH]** WHATWG, "Fetch Standard", December 2021, <https://fetch.spec.whatwg.org/>.

**[PERF]** Sy, E., Moennich, M., Mueller, T., Federrath, H., and M. Fischer, "Enhanced Performance for the encrypted Web through TLS Resumption across Hostnames", 7 February 2019, <https://arxiv.org/pdf/1902.02531.pdf>.

## Acknowledgments

Cross-name resumption has been previously implemented in the QUIC Crypto protocol as a preloaded list of hostnames.

Erik Sy has previously proposed a similar mechanism for TLS, draft-sy-tls-resumption-group. This document incorporates ideas from that draft.

This document has benefited from contributions and suggestions from Carrick Bartle, David Benjamin, Nick Harper, Eric Rescorla, David Schinazi, Ryan Sleevi, Ian Swett, Martin Thomson, Christopher Wood, and many others.

## Author's Address

Victor Vasiliev
Google

Email: vasilvv@google.com