Network Working Group
Internet-Draft
Updates: 2246,4346,5246
(if approved)
Intended status: Standards Track
Expires: December 3, 2014

B. Moeller
A. Langley
Google

June 1, 2014

# TLS Fallback Signaling Cipher Suite Value (SCSV) for Preventing Protocol Downgrade Attacks

draft-bmoeller-tls-downgrade-scsv-02

## Abstract

This document defines a Signaling Cipher Suite Value (SCSV) that prevents protocol downgrade attacks on the Transport Layer Security (TLS) protocol. It updates RFC 2246, RFC 4346, and RFC 5246.

## Status of this Memo

## Copyright Notice

Table of Contents

## 1. Introduction

To work around interoperability problems with legacy servers, many TLS client implementations do not rely on the TLS protocol version negotiation mechanism alone, but will intentionally reconnect using a downgraded protocol if initial handshake attempts fail. Such clients may fall back to connections in which they announce a version as low as TLS 1.0 (or even its predecessor, SSL 3.0) as the highest supported version.

While such protocol downgrades can be a useful last resort for connections to actual legacy servers, there's a risk that active attackers could exploit the downgrade strategy to weaken the cryptographic security of connections. Also, handshake errors due to network glitches could similarly be misinterpreted as interaction with a legacy server and result in a protocol downgrade.

All unnecessary protocol downgrades are undesirable (e.g., from TLS 1.2 to TLS 1.1 if both the client and the server actually do support TLS 1.2); they can be particularly critical if they mean losing the TLS extension feature (when downgrading to SSL 3.0). This document defines a Signaling Cipher Suite Value (SCSV) that can be employed to prevent unintended protocol downgrades between clients and servers that comply to this document, by having the client indicate that the current connection attempt is merely a fallback.

This specification applies to implementations of TLS 1.0 [RFC2246], TLS 1.1 [RFC4346], and TLS 1.2 [RFC5246]. (It is particularly relevant if such implementations also include support for predecessor protocol SSL 3.0 [RFC6101].) It can be applied similarly to later protocol versions.

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 [RFC2119].

## 2. Protocol values

This document defines a new TLS cipher suite value:

```
TLS_FALLBACK_SCSV          {0x56, 0x00}
```

This is a signaling cipher suite value (SCSV), i.e., it does not actually correspond to a suite of cryptosystems, and it can never be selected by the server in the handshake; rather, its presence in the client hello message serves as a backwards-compatible signal from the client to the server.

This document also allocates a new alert value in the TLS Alert Registry [RFC5246]:

```
enum {
  /* ... */
  inappropriate_fallback(86),
  /* ... */
  (255)
} AlertDescription;
```

This alert is only generated by servers, as described in Section 3. It is always fatal.

## 3. Server behavior

This section specifies server behavior when receiving the TLS_FALLBACK_SCSV cipher suite from a client in ClientHello.cipher_suites.

- If TLS_FALLBACK_SCSV appears in ClientHello.cipher_suites and the highest protocol version supported by the server is higher than the version indicated in ClientHello.client_version, the server MUST respond with an inappropriate_fallback alert.

Otherwise (either TLS_FALLBACK_SCSV does not appear, or it appears and the client's protocol version is at least the highest protocol version supported by the server), the server proceeds with the handshake as usual.

## 4. Client behavior

The TLS_FALLBACK_SCSV cipher suite value is meant for use by clients that repeat a connection attempt with a downgraded protocol in order to avoid interoperability problems with legacy servers. This section specifies when to send it.

- If a client sends a ClientHello.client_version containing a lower value than the latest (highest-valued) version supported by the client, it SHOULD include the TLS_FALLBACK_SCSV cipher suite value in ClientHello.cipher_suites. (Since the cipher suite list in the ClientHello is ordered by preference, with the client's favorite choice first, signaling cipher suite values will generally appear after all cipher suites that the client actually intends to negotiate.)

  However, as an exception to the above, when the client intends to perform an abbreviated handshake to resume a previously negotiated session and sets ClientHello.client_version to the protocol version negotiated for that session, the client MUST NOT include TLS_FALLBACK_SCSV in ClientHello.cipher_suites.

Note that in the above, a protocol version is not considered supported by the client if it has been disabled by any applicable system or user settings: it is about the highest protocol version that the client would attempt using in a handshake, not about the highest protocol version implemented if its use is not actually enabled. (For example, if the implementation supports TLS 1.2 but the user has disabled this protocol version, a TLS 1.1 handshake is expected and does not warrant sending TLS_FALLBACK_SCSV.)

## 5. Security Considerations

Section 4 does not require client implementations to send TLS_FALLBACK_SCSV in any particular case, it merely recommends it; behavior can be adapted according to the client's security needs. For example, during the initial deployment of a new protocol version (when some interoperability problems may have to be expected), smoothly falling back to the previous protocol version in case of problems may be preferrable to potentially not being able to connect at all: so TLS_FALLBACK_SCSV could be omitted for this particular protocol downgrade step.

However, it is strongly recommended to send TLS_FALLBACK_SCSV when downgrading to SSL 3.0 as the CBC cipher suites in SSL 3.0 have weaknesses that cannot be addressed by implementation workarounds like the remaining weaknesses in later (TLS) protocol versions.

## 6. IANA Considerations

[[ NOTE IN DRAFT: The requested registry allocation requires Standards Action, i.e., will only be official with the IESG's Standards Track RFC approval. Since this document is currently an Internet-Draft, IANA so far has in fact not added the cipher suite number to the registry. ]]

IANA has added TLS cipher suite number 0x56,0x00 with name TLS_FALLBACK_SCSV to the TLS Cipher Suite registry.

# 7. References

## 7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

## 7.2. Informal References

[RFC6101] Freier, A., Karlton, P., and P. Kocher, "The Secure Sockets Layer (SSL) Protocol Version 3.0", RFC 6101, August 2011.

## Appendix A.  Acknowledgements

This specification was inspired by an earlier proposal by Eric Rescorla.

## Authors' Addresses

Bodo Moeller
Google Switzerland GmbH
Brandschenkestrasse 110
Zurich  8002
Switzerland

Email: bmoeller@acm.org

Adam Langley
Google Inc.
345 Spear St
San Francisco, CA  94105
USA

Email: agl@google.com