

Network Working Group
Internet-Draft
Updates: [2246](#),4346,5246
(if approved)
Intended status: Standards Track
Expires: March 29, 2014

B. Moeller
Google
September 25, 2013

TLS Signaling Cipher Suite Value (SCSV) for Preventing Protocol
Downgrade Attacks
draft-bmoeller-tls-downgrade-scsv-00

Abstract

This document defines a Signaling Cipher Suite Value (SCSV) that can be used to prevent protocol downgrades for Transport Layer Security (TLS).

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 <http://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 March 29, 2014.

Copyright Notice

Copyright (c) 2013 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 (<http://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

Internet-Draft

TLS Downgrade SCSV

September 2013

described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	The TLS_DOWNGRADE_SCSV Signaling Cipher Suite Value	4
3.	Server behavior	5
4.	Client behavior	6
5.	Security Considerations	7
6.	References	8
6.1.	Normative References	8
6.2.	Informal References	8
Appendix A.	Acknowledgements	9
	Author's Address	10

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, and make no attempt of using TLS extensions.

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. (For example, if a server requires the client's Supported Elliptic Curves Extension [[RFC4492](#)] to choose a forward-secure key exchange algorithm, the attacker could interfere with connections using this extension to get the client and server to instead use a key exchange algorithm that does not providing forward secrecy.) Also, handshake errors due to network glitches could similiary 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 1.1); they can be particularly critical if they mean losing the TLS extension feature (when downgrading to TLS 1.0 without extensions, or to SSL 3.0). This document defines a Signaling Cipher Suite Value (SCSV) that can be employed to prevent such protocol downgrades between clients and servers that comply to this document.

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.](#) The TLS_DOWNGRADE_SCSV Signaling Cipher Suite Value

This document defines one new cipher suite value:

TLS_DOWNGRADE_SCSV {0x##, 0x##}

[[Code point TBD.]]

This is a signaling cipher suite value, 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.

[3.](#) Server behavior

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

- o If TLS_DOWNGRADE_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 a (fatal) protocol_version alert.
- o If TLS_DOWNGRADE_SCSV appears in ClientHello.cipher_suites and the ClientHello message does not include field client_hello_extension_list at all ([\[RFC3546\]](#), [\[RFC4366\]](#), [\[RFC5246\]](#)), the server MUST respond with a (fatal) protocol_version alert.

Otherwise (either TLS_DOWNGRADE_SCSV does not appear, or it appears *and* the client's protocol version is at least the highest protocol version supported by the server *and* there is a client_hello_extension_list - possibly empty), the server proceeds with the handshake as usual.

(Note that the TLS specifications require server implementations that do not support TLS extensions to tolerate extra data at the end of the ClientHello message, at the location where the later specifications added the field `client_hello_extension_list`. The server behavior mandated here relies on that requirement; intolerance for such extra data is a bug that must be fixed before adding server-side support for TLS_DOWNGRADE_SCSV.)

[4.](#) Client behavior

The TLS_DOWNGRADE_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.

- o 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_DOWNGRADE_SCSV cipher suite value in `ClientHello.cipher_suites`. This does not apply 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.

- o If a client that supports TLS extensions and has TLS extensions to send in the handshake entirely omits `client_hello_extension_list` from the `ClientHello` message, it SHOULD include the `TLS_DOWNGRADE_SCSV` cipher suite value 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_DOWNGRADE_SCSV`.)

[5.](#) Security Considerations

[Section 4](#) does not require client implementations to send `TLS_DOWNGRADE_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 preferable to potentially not being able to connect at all: so TLS_DOWNGRADE_SCSV could be omitted for this particular protocol downgrade step.

However, it is particularly strongly recommended to send TLS_DOWNGRADE_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.

6.1. 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.
- [RFC3546] Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., and T. Wright, "Transport Layer Security (TLS) Extensions", [RFC 3546](#), June 2003.
- [RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", [RFC 4346](#), April 2006.
- [RFC4366] Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., and T. Wright, "Transport Layer Security (TLS) Extensions", [RFC 4366](#), April 2006.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), August 2008.

6.2. Informal References

- [RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", [RFC 4492](#), May 2006.
- [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.

Internet-Draft

TLS Downgrade SCSV

September 2013

Author's Address

Bodo Moeller
Google Switzerland GmbH
Brandschenkestrasse 110
Zurich 8002
Switzerland

Email: bmoeller@acm.org

Moeller

Expires March 29, 2014

[Page 10]