Abstract

This document, if approved, formally deprecates Transport Layer Security (TLS) versions 1.0 [RFC2246] and 1.1 [RFC4346] and moves these documents to the historic state. These versions lack support for current and recommended cipher suites, and various government and industry profiles of applications using TLS now mandate avoiding these old TLS versions. TLSv1.2 has been the recommended version for IETF protocols since 2008, providing sufficient time to transition away from older versions. Products having to support older versions increase the attack surface unnecessarily and increase opportunities for misconfigurations. Supporting these older versions also requires additional effort for library and product maintenance.

This document also deprecates Datagram TLS (DTLS) version 1.0 [RFC6347] (but not DTLS version 1.2, and there is no DTLS version 1.1).

This document updates many RFCs that normatively refer to TLSv1.0 or TLSv1.1 as described herein. This document also updates RFC 7525 and hence is part of BCP195.
1. Introduction

Transport Layer Security (TLS) versions 1.0 [RFC2246] and 1.1 [RFC4346] were superceded by TLSv1.2 [RFC5246] in 2008, which has now itself been superceded by TLSv1.3 [RFC8446]. It is therefore timely to further deprecate these old versions. The expectation is that TLSv1.2 will continue to be used for many years alongside TLSv1.3.

The primary technical reasons for deprecating these versions include:

- They require implementation of older cipher suites that are no longer desirable for cryptographic reasons, e.g. TLSv1.0 makes TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA mandatory to implement
- Lack of support for current recommended cipher suites, especially using AEAD ciphers which are not supported prior to TLSv1.2.
  Note: registry entries for no-longer-desirable ciphersuites remain in the registries, but many TLS registries are being updated through [RFC8447] which denotes such entries as "not recommended."
- Integrity of the handshake depends on SHA-1 hash
- Authentication of the peers depends on SHA-1 signatures
- Support for four protocol versions increases the likelihood of misconfiguration
- At least one widely-used library has plans to drop TLSv1.1 and TLSv1.0 support in upcoming releases; products using such libraries would need to use older versions of the libraries to support TLSv1.0 and TLSv1.1, which is clearly undesirable

Deprecation of these versions is intended to assist developers as additional justification to no longer support older TLS versions and to migrate to a minimum of TLSv1.2. Deprecation also assists product teams with phasing out support for the older versions to reduce the attack surface and the scope of maintenance for protocols in their offerings.

1.1. RFCs Updated

This document updates the following RFCs that normatively reference TLSv1.0 or TLSv1.1 or DTLS1.0. The update is to obsolete usage of these older versions. Fallback to these versions are prohibited through this update.

[RFC8465] [RFC8422] [RFC8261] [RFC7568] [RFC7562] [RFC7525] [RFC7507] [RFC7465] [RFC6750] [RFC6749] [RFC6739] [RFC6460] [RFC6084] [RFC6083] [RFC6367] [RFC6176] [RFC6042] [RFC6012] [RFC5878] [RFC5734] [RFC5469] [RFC5456] [RFC5422] [RFC5415] [RFC5364] [RFC5281] [RFC5263] [RFC5238]
In addition these RFCs normatively refer to TLSv1.0 or TLSv1.1 and have been obsoleted: [RFC5101] [RFC5081] [RFC5077] [RFC4934] [RFC4572] [RFC4507] [RFC4492] [RFC4366] [RFC4347] [RFC4244] [RFC4132] [RFC3920] [RFC3734] [RFC3588] [RFC3546] [RFC3489] [RFC3316]

In the case of [RFC4642], that has already been updated by [RFC8143] which makes an overlapping, but not quite the same, update as this document.

This document updates DTLS [RFC6347].

1.2. Terminology

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.

2. Support for Deprecation

Specific details on attacks against TLSv1.0 and TLSv1.1 as well as their mitigations are provided in NIST SP800-52r2 [NIST800-52r2], RFC 7457 [RFC7457] and other referenced RFCs. Although the attacks have been mitigated, if support is dropped for future library releases for these versions, it is unlikely attacks found going forward will be mitigated in older library releases.

NIST for example have provided the following rationale, copied with permission from NIST SP800-52r2 [NIST800-52r2], section 1.2 "History of TLS" (with references changed for RFC formatting).

TLS 1.1, specified in [RFC4346], was developed to address weaknesses discovered in TLS 1.0, primarily in the areas of initialization vector selection and padding error processing. Initialization vectors were made explicit to prevent a certain class of attacks on the Cipher Block Chaining (CBC) mode of operation used by TLS. The handling of padding errors was altered to treat a padding error as a bad message authentication code,
rather than a decryption failure. In addition, the TLS 1.1 RFC acknowledges attacks on CBC mode that rely on the time to compute the message authentication code (MAC). The TLS 1.1 specification states that to defend against such attacks, an implementation must process records in the same manner regardless of whether padding errors exist. Further implementation considerations for CBC modes (which were not included in RFC4346 [RFC4346]) are discussed in Section 3.3.2.

TLSv1.2, specified in RFC5246 [RFC5246], made several cryptographic enhancements, particularly in the area of hash functions, with the ability to use or specify the SHA-2 family algorithms for hash, MAC, and Pseudorandom Function (PRF) computations. TLSv1.2 also adds authenticated encryption with associated data (AEAD) cipher suites.

TLS 1.3, specified in TLSv1.3 [RFC8446], represents a significant change to TLS that aims to address threats that have arisen over the years. Among the changes are a new handshake protocol, a new key derivation process that uses the HMAC-based Extract-and-Expand Key Derivation Function (HKDF), and the removal of cipher suites that use static RSA or DH key exchanges, the CBC mode of operation, or SHA-1. The list of extensions that can be used with TLS 1.3 has been reduced considerably.

3. SHA-1 Usage Problematic in TLSv1.0 and TLSv1.1

The integrity of both TLSv1.0 and TLSv1.1 depends on a running SHA-1 hash of the exchanged messages. This makes it possible to perform a downgrade attack on the handshake by an attacker able to perform 2^77 operations, well below the acceptable modern security margin.

Similarly, the authentication of the handshake depends on signatures made using SHA-1 hash or a not stronger concatenation of MD-5 and SHA-1 hashes, allowing the attacker to impersonate a server when it is able to break the severely weakened SHA-1 hash.

Neither TLSv1.0 nor TLSv1.1 allow the peers to select a stronger hash for signatures in the ServerKeyExchange or CertificateVerify messages, making the only upgrade path the use of a newer protocol version.

See [Bhargavan2016] for additional detail.
4. Do Not Use TLSv1.0

TLSv1.0 MUST NOT be used. Negotiation of TLSv1.0 from any version of TLS MUST NOT be permitted.

Any other version of TLS is more secure than TLSv1.0. TLSv1.0 can be configured to prevent interception, though using the highest version available is preferable.

Pragmatically, clients MUST NOT send a ClientHello with ClientHello.client_version set to \{03,01\}. Similarly, servers MUST NOT send a ServerHello with ServerHello.server_version set to \{03,01\}. Any party receiving a Hello message with the protocol version set to \{03,01\} MUST respond with a "protocol_version" alert message and close the connection.

Historically, TLS specifications were not clear on what the record layer version number (TLSPlaintext.version) could contain when sending ClientHello. [RFC5246] notes that TLSPlaintext.version could be selected to maximize interoperability, though no definitive value is identified as ideal. That guidance is still applicable; therefore, TLS servers MUST accept any value \{03,XX\} (including \{03,00\}) as the record layer version number for ClientHello, but they MUST NOT negotiate TLSv1.0.

5. Do Not Use TLSv1.1

TLSv1.1 MUST NOT be used. Negotiation of TLSv1.1 from any version of TLS MUST NOT be permitted.

Pragmatically, clients MUST NOT send a ClientHello with ClientHello.client_version set to \{03,02\}. Similarly, servers MUST NOT send a ServerHello with ServerHello.server_version set to \{03,02\}. Any party receiving a Hello message with the protocol version set to \{03,02\} MUST respond with a "protocol_version" alert message and close the connection.

Any newer version of TLS is more secure than TLSv1.1. TLSv1.1 can be configured to prevent interception, though using the highest version available is preferable. Support for TLSv1.1 is dwindling in libraries and will impact security going forward if mitigations for attacks cannot be easily addressed and supported in older libraries.

Historically, TLS specifications were not clear on what the record layer version number (TLSPlaintext.version) could contain when sending ClientHello. [RFC5246] notes that TLSPlaintext.version could be selected to maximize interoperability, though no definitive value is identified as ideal. That guidance is
still applicable; therefore, TLS servers MUST accept any value
{03,XX} (including {03,00}) as the record layer version number for
ClientHello, but they MUST NOT negotiate TLSv1.1.

6. Updates to RFC7525

RFC7525 is BCP195, "Recommendations for Secure Use of Transport Layer
Security (TLS) and Datagram Transport Layer Security (DTLS)", is the
most recent best practice document for implementing TLS and was based
on TLSv1.2. At the time of publication, TLSv1.0 and TLSv1.1 had not
yet been deprecated. As such, this document is called out
specifically to update text implementing the deprecation
recommendations of this document.

This documents updates [RFC7525] Section 3.1.1 changing SHOULD NOT to
MUST NOT as follows:

- Implementations MUST NOT negotiate TLS version 1.0 [RFC2246].
  Rationale: TLSv1.0 (published in 1999) does not support many
  modern, strong cipher suites. In addition, TLSv1.0 lacks a per-
  record Initialization Vector (IV) for CBC-based cipher suites and
  does not warn against common padding errors.

- Implementations MUST NOT negotiate TLS version 1.1 [RFC4346].
  Rationale: TLSv1.1 (published in 2006) is a security improvement
  over TLSv1.0 but still does not support certain stronger cipher
  suites.

This documents updates [RFC7525] Section 3.1.2 changing SHOULD NOT to
MUST NOT as follows:

- Implementations MUST NOT negotiate DTLS version 1.0 [RFC4347],
  [RFC6347].
  Version 1.0 of DTLS correlates to version 1.1 of TLS (see above).

7. Security Considerations

This document deprecates two older protocol versions for security
reasons already described. The attack surface is reduced when there
are a smaller number of supported protocols and fallback options are
removed.
8. Acknowledgements

Thanks to those that provided usage data, reviewed and/or improved
this document, including: David Benjamin, David Black, Viktor
Dukhovni, Julien Elie, Gary Gapinski, Alessandro Ghedini, Jeremy
Harris, Russ Housley, Hubert Kario, John Mattsson, Eric Mill, Yoav
Nir, Andrei Popov, Eric Rescorla, Yaron Sheffer, Robert Sparks,
Martin Thomson, Loganaden Velvindron, https://github.com/yaleman, and
Jakub Wilk.

[[Note to RFC editor: At least Julien Elie's name above should have
an accent on the first letter of the surname. Please fix that and
any others needing a similar fix if you can, I'm not sure the tooling
I have now allows that.]]

9. IANA Considerations

[[This memo includes no request to IANA.]]

10. References

10.1. 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,

RFC 2246, DOI 10.17487/RFC2246, January 1999,
<https://www.rfc-editor.org/info/rfc2246>.

[RFC3261]  Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston,
A., Peterson, J., Sparks, R., Handley, M., and E.
Schooler, "SIP: Session Initiation Protocol",
RFC 3261, DOI 10.17487/RFC3261, June 2002,

Haukka, "Security Mechanism Agreement for the Session
Initiation Protocol (SIP)", RFC 3329,
DOI 10.17487/RFC3329, January 2003,

Layer Security over Stream Control Transmission Protocol",
RFC 3436, DOI 10.17487/RFC3436, December 2002,


10.2. Informative References


Internet-Draft       Deprecating TLSv1.0 and TLSv1.1           June 2019


Appendix A. Change Log

[[RFC editor: please remove this before publication.]]

From draft-ietf-tls-oldversions-deprecate-04 to draft-ietf-tls-oldversions-deprecate-05:

- Removed references to government related deprecation statements: US, Canada, and Germany. NIST documentation rationale remains as a reference describing the relevant RFCs and justification.

From draft-ietf-tls-oldversions-deprecate-02 to draft-ietf-tls-oldversions-deprecate-03:

- Added 8261 to updates list based on IETF-104 meeting.

From draft-ietf-tls-oldversions-deprecate-01 to draft-ietf-tls-oldversions-deprecate-02:

- Correction: 2nd list of referenced RFCs in Section 1.1 aren't informatively referring to tls1.0/1.1
- Remove RFC7255 from updates list - datatracker has bad data (spotted by Robert Sparks)
- Added point about RFCs 8143 and 4642
- Added UPDATEs for RFCs that refer to 4347 and aren't OBSOLETED
- Added note about RFC8261 to see what WG want.

From draft-ietf-tls-oldversions-deprecate-00 to draft-ietf-tls-oldversions-deprecate-01:

- PRs with typos and similar: so far just #1
- PR#2 noting msft browser announced deprecation (but this was OBE as per...)
- Implemented actions as per IETF-103 meeting:
  * Details about which RFC's, BCP's are affected were generated using a script in the git repo: https://github.com/tlswg/oldversions-deprecate/blob/master/nonobsnorms.sh
  * Removed the 'measurements' part
  * Removed SHA-1 deprecation (section 8 of -00)

From draft-moriarty-tls-oldversions-diediedie-01 to draft-ietf-tls-oldversions-deprecate-00:

- I-Ds became RFCs 8446/8447 (old-repo PR#4, for TLSv1.3)
- Accepted old-repo PR#5 fixing typos
From draft-moriarty-tls-oldversions-diediedie-00 to draft-moriarty-tls-oldversions-diediedie-01:

- Added stats sent to list so far
- PR's #2,3
- A few more references
- Added section on email

Authors' Addresses

Kathleen Moriarty
Dell EMC
176 South Street
Hopkinton
United States

EMail: Kathleen.Moriarty.ietf@gmail.com

Stephen Farrell
Trinity College Dublin
Dublin 2
Ireland

Phone: +353-1-896-2354
EMail: stephen.farrell@cs.tcd.ie