Abstract

This document, if approved, formally deprecates Transport Layer Security (TLS) versions 1.0 (RFC 2246) and 1.1 (RFC 4346). Accordingly, those documents (will be moved|have been moved) to Historic status. These versions lack support for current and recommended cryptographic algorithms and mechanisms, 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. Removing support for older versions from implementations reduces the attack surface, reduces opportunity for misconfiguration, and streamlines library and product maintenance.

This document also deprecates Datagram TLS (DTLS) version 1.0 (RFC4347), 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 the best practices for TLS usage in RFC 7525 and hence is part of BCP195.
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 June 18, 2021.

Copyright Notice

Copyright (c) 2020 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 ................................................. 3
   1.1. RFCs Updated ........................................... 3
   1.2. Terminology ............................................ 5
2. Support for Deprecation ..................................... 5
3. SHA-1 Usage Problematic in TLSv1.0 and TLSv1.1 ............... 6
4. Do Not Use TLSv1.0 .......................................... 6
5. Do Not Use TLSv1.1 .......................................... 7
6. Updates to RFC7525 .......................................... 8
7. Operational Considerations ................................ 8
8. Security Considerations .................................... 9
9. Acknowledgements .......................................... 9
10. IANA Considerations ....................................... 9
11. References ................................................. 9
   11.1. Normative References ................................... 9
1. Introduction

Transport Layer Security (TLS) versions 1.0 [RFC2246] and 1.1 [RFC4346] were superseded by TLSv1.2 [RFC5246] in 2008, which has now itself been superseded by TLSv1.3 [RFC8446]. Datagram Transport Layer Security (DTLS) version 1.0 [RFC4347] was superseded by DTLSv1.2 [RFC6347] in 2012. It is therefore timely to further deprecate these old versions. Accordingly, those documents (will be moved|have been moved) to Historic status.

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 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 indicates that such entries are not recommended by the IETF.
- Integrity of the handshake depends on SHA-1 hash.
- Authentication of the peers depends on SHA-1 signatures.
- Support for four TLS 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 (D)TLS versions and to migrate to a minimum of (D)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. Specific references to mandatory minimum protocol versions of TLSv1.0 or TLSv1.1 are replaced by TLSv1.2, and
references to minimum protocol version DTLSv1.0 are replaced by DTLSv1.2. Statements that "TLSv1.0 is the most widely deployed version and will provide the broadest interoperability" are removed without replacement.

The status of [RFC7562], [RFC6042], [RFC5456], [RFC5024], [RFC4540], and [RFC3656] will be updated with permission of the Independent Stream Editor.

In addition these RFCs normatively refer to TLSv1.0 or TLSv1.1 and have already been obsoleted; they are still listed here and marked as updated by this document in order to reiterate that any usage of the obsolete protocol should still use modern TLS: [RFC5953] [RFC5101] [RFC5081] [RFC5077] [RFC4934] [RFC4572] [RFC4507] [RFC4507] [RFC4492] [RFC4366] [RFC4347] [RFC4244] [RFC4132] [RFC3920] [RFC3734] [RFC3588] [RFC3546] [RFC3489] [RFC3316]

Note that [RFC4642] has already been updated by [RFC8143], which makes an overlapping, but not quite identical, update as this document.

[RFC6614] has a requirement for TLSv1.1 or later, although only makes an informative reference to [RFC4346]. This requirement is updated to be for TLSv1.2 or later.

[RFC6460], [RFC4744], and [RFC4743] are already Historic; they are still listed here and marked as updated by this document in order to reiterate that any usage of the obsolete protocol should still use modern TLS.

This document updates DTLS [RFC6347]. [RFC6347] had allowed for negotiating the use of DTLSv1.0, which is now forbidden.

The DES and IDEA cipher suites specified in [RFC5469] were specifically removed from TLSv1.2 by [RFC5246]; since the only
versions of TLS for which their usage is defined are now Historic, RFC 5469 (will be|has been) moved to Historic as well.

The version-fallback Signaling Cipher Suite Value specified in [RFC7507] was defined to detect when a given client and server negotiate a lower version of (D)TLS than their highest shared version. TLSv1.3 ([RFC8446]) incorporates a different mechanism that achieves this purpose, via sentinel values in the ServerHello.Random field. With (D)TLS versions prior to 1.2 fully deprecated, the only way for (D)TLS implementations to negotiate a lower version than their highest shared version would be to negotiate (D)TLSv1.2 while supporting (D)TLSv1.3; supporting (D)TLSv1.3 implies support for the ServerHello.Random mechanism. Accordingly, the functionality from [RFC7507] has been superseded, and this document marks it as Obsolete.

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 [NIST800-52r2], RFC 7457 [RFC7457] and other RFCs referenced therein. Although mitigations for the current known vulnerabilities have been developed, any future issues discovered in old protocol versions might not be mitigated in older library versions when newer library versions do not support those old protocols.

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

TLSv1.1, specified in [RFC4346], was developed to address weaknesses discovered in TLSv1.0, 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 TLSv1.1 RFC acknowledges attacks on CBC mode that rely on the time to compute the message authentication code (MAC). The TLSv1.1 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.

TLSv1.3, 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 TLSv1.3 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 a SHA-1 hash or a not appreciably 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. While TLSv1.0 can be configured to prevent some types of interception, using the highest version available is preferred.

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. Appendix E of [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. While TLSv1.1 can be configured to prevent some types of interception, using the highest version available is preferred. 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. Appendix E of [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)", which 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, BCP195 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. Operational Considerations

This document is part of BCP 195, and as such reflects the understanding of the IETF (at the time of its publication) as to the best practices for TLS and DTLS usage.

Though TLSv1.1 has been obsolete since the publication of RFC 5246 in 2008, and DTLSv1.0 has been obsolete since the publication of RFC 6347 in 2012, there may remain some systems in operation that do not support (D)TLSv1.2 or higher. Adopting the practices recommended by this document for any systems that need to communicate with the aforementioned class of systems will cause failure to interoperate. However, disregarding the recommendations of this document in order to continue to interoperate with the aforementioned class of systems
incurs some amount of risk. The nature of the risks incurred by operating in contravention to the recommendations of this document are discussed in Sections 2 and 3, and knowledge of those risks should be used along with any potential mitigating factors and the risks inherent to updating the systems in question when deciding how quickly to adopt the recommendations specified in this document.

8. Security Considerations

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

9. Acknowledgements

Thanks to those that provided usage data, reviewed and/or improved this document, including: Michael Ackermann, David Benjamin, David Black, Deborah Brungard, Alan DeKok, Viktor Dukhovni, Julien Elie, Adrian Farrell, Gary Gapinski, Alessandro Ghedini, Peter Gutmann, Jeremy Harris, Nick Hilliard, James Hodgkinson, Russ Housley, Hubert Kario, Benjamin Kaduk, John Klensin, Watson Ladd, Eliot Lear, Ted Lemon, John Mattsson, Keith Moore, Tom Petch, Eric Mill, Yoav Nir, Andrei Popov, Michael Richardson, Eric Rescorla, Rich Salz, Mohit Sethi, Yaron Sheffer, Rob Sayre, Robert Sparks, Barbara Stark, Martin Thomson, Sean Turner, Loganaden Velvindron, 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.]]

10. IANA Considerations

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

11. References

11.1. Normative References


11.2. Informative References


Appendix A. Change Log

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

From draft-ietf-tls-oldversions-deprecate-10 to draft-ietf-tls-oldversions-deprecate-11:

- RFC 5953 was mentioned in the wrong para of section 1.1 - it has been obsoleted already.

From draft-ietf-tls-oldversions-deprecate-09 to draft-ietf-tls-oldversions-deprecate-10:

- We missed adding change logs for a few versions, but since -09 was the one that underwent IETF last call, and there was some discussion, we figured it'd be good to mention substantive changes here.
- Added Ben's suggested text for "operational considerations" following extensive last call discussion.
- Re-checked the references to RFC 4347 after Tom Petch noticed we missed a couple. Added RFCs 5953 and 6353 to the list here. All others were in already.
- Fixed various typos and ack'd those who engaged a bit in the IETF LC discussion. (If we missed you and you want to be added, or if you'd rather not be mentioned, just ping the authors.)

From draft-ietf-tls-oldversions-deprecate-05 to draft-ietf-tls-oldversions-deprecate-06:

- Fixed "yaleman" ack.
- Added RFC6614 to UPDATEs list.
- per preliminary AD review:
  * Remove references from abstract
  * s/primary technical reasons/technical reasons/
  * Add rfc7030 to 1.1
  * verified that all the RFCs in the (massive:-) Updates meta-data are mentioned in section 1.1 (I think appropriately;-)

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

- Removed references to gouverment related deprecation statements: US, Canada, and Germany. NIST documentation rationale remains as a reference describing the relevent 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 refering 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