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 updates many RFCs that normatively refer to TLS1.0 or TLS1.1 as described herein. This document also updates 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 May 12, 2019.

Copyright Notice

Copyright (c) 2018 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. Updates ......................................... 3
   1.2. Terminology ..................................... 4
2. Support for Deprecation ................................. 4
3. SHA-1 .................................................. 5
4. Do Not Use TLSv1.0 ..................................... 6
5. Do Not Use TLSv1.1 ..................................... 6
6. Updates to RFC7525 ..................................... 7
7. Security Considerations ................................. 7
8. Acknowledgements ....................................... 7
9. IANA Considerations .................................... 8
10. References ........................................... 8
   10.1. Normative References .............................. 8
   10.2. Informative References ........................... 16
Appendix A. Change Log .................................... 20
Authors’ Addresses ....................................... 20
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]. 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.

TLSv1.1 and TLSv1.0 are also actively being deprecated in accordance with guidance from government agencies (e.g. NIST SP 80052r2 [NIST800-52r2]) and industry consortia such as the Payment Card Industry Association (PCI) [PCI-TLS1].

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 TLS 1.2. 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. Updates

This document updates these RFCs that normatively reference TLS1.0 or TLS1.1 and have not been obsoleted: [RFC8465] [RFC8422] [RFC7568] [RFC7562] [RFC7507] [RFC7465] [RFC7255] [RFC7030] [RFC6750] [RFC6749] [RFC6739] [RFC6367] [RFC6176] [RFC6042] [RFC5878] [RFC5734] [RFC5469] [RFC5422] [RFC5364] [RFC5281] [RFC5263] [RFC5238] [RFC5216] [RFC5158] [RFC5091] [RFC5054] [RFC5049] [RFC5024] [RFC5023] [RFC5019] [RFC5018] [RFC4992] [RFC4976] [RFC4975] [RFC4964] [RFC4851] [RFC4823] [RFC4791]
In addition these RFCs normatively refer to TLS1.0 or TLS1.1 and have
been obsoleted, or informatively refer to TLS1.0 or TLS1.1: [RFC5101]
[RFC5081] [RFC5077] [RFC4934] [RFC4572] [RFC4507] [RFC4492] [RFC4366]
[RFC4347] [RFC4244] [RFC4132] [RFC3920] [RFC3734] [RFC3588] [RFC3546]
[RFC3489] [RFC3316]

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
[RFCC2119] [RFC8174] when, and only when, they appear in all
capitals, as shown here.

2. Support for Deprecation

Industry has actively followed guidance provided by NIST and the PCI
Council to deprecate TLSv1.0 and TLSv1.1 by June 30, 2018. TLSv1.2
should remain a minimum baseline for TLS support at this time.

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.

TLS 1.2, 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. TLS 1.2 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.

The Canadian government treasury board have also mandated that these old versions of TLS not be used. [Canada]

Various companies and web sites have announced plans to deprecate these old versions of TLS.

3. SHA-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. 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. 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. 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

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: TLS 1.0 (published in 1999) does not support many modern, strong cipher suites. In addition, TLS 1.0 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: TLS 1.1 (published in 2006) is a security improvement over TLS 1.0 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].
  
  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, Alessandro Ghedini, Jeremy Harris, Russ Housley, Hubert Kario, Loganaden Velvindron, Eric Mill, Yoav Nir, Andrei Popov, Eric Rescorla, Yaron Sheffer, https://github.com/yaleman, and Jakub Wilk.
9. IANA Considerations

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

10. References

10.1. Normative References


Internet-Draft       Deprecating TLSv1.0 and TLSv1.1       November 2018


10.2. Informative References


Appendix A. Change Log

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

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 TLS1.3)
- 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