Deprecating TLSv1.0 and TLSv1.1
draft-ietf-tls-oldversions-deprecate-00

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 the backward compatibility sections of TLS RFCs
   [[list TBD]] to prohibit fallback to TLSv1.0 and TLSv1.1. This
   document also updates RFC 7525.

Copyright Notice

Table of Contents

1. Introduction

   [[Text in double-square brackets is intended to be fixed as the draft
   evolves. You've seen that we need to figure out the list of RFCs
   that this'd update in the abstract. There is a repo for this at:
   https://github.com/tlswg/oldversions-deprecate - PRs (on the xml
   file) are welcome there.]]

Transport Layer Security (TLS) versions 1.0 [RFC2246] and 1.1 [RFC4346] were superceded by TLSv1.2 [RFC5246] in 2008, which has now itself been superceded by TLSv1.3 [RFC8446].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.

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:

o 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
o 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."
o Integrity of the handshake depends on SHA-1 hash
o Authentication of the peers depends on SHA-1 signatures
o Support for four protocol versions increases the likelihood of
   misconfiguration
o 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.

[[This draft is being written now so that the TLS WG chairs can just hit the "publication requested" button as soon as there is WG consensus to deprecate these ancient versions of TLS. The authors however think that deprecation now is timely.]]

1.1. 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

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 mandated that these old
versions of TLS not be used. [Canada]

3. Removing Support

[[This section can be removed upon publication - or maybe keep it?]]

Support for TLSv1.0 has been removed by the July 2018 PCI deadline
from the following standards, products, and services:

o 3GPP 5G
o Amazon Elastic Load Balancing [Amazon]
o CloudFlare [CloudFlare]
o Digicert [Digicert]
o GitHub [GIT]
o KeyCDN [KeyCDN]
o PayPal [paypal]
o Stripe [stripe]
o [[Numerous web sites...]]

Many web sites have taken the action of including the deprecation of
TLSv1.1 into their plans for deprecating TLSv1.0 for the PCI council
deadline. Support for TLSv1.1 has been removed by the July 2018 PCI
deadline from the following standards, products, and services:

o 3GPP 5G Release 16
o Amazon Elastic Load Balancing [Amazon]
o CloudFlare [CloudFlare]
o GitHub [GIT]
o PayPal [paypal]
o Stripe [stripe]
o [[Numerous web sites...]]

4. Usage

[[This section can be removed upon publication - or maybe keep it?]]

4.1. Web

Usage statistics for TLSv1.0 and TLSv1.1 on the public web vary, but
have been in general very low and declined further with the impending
PCI deadline to migrate off of TLSv1.0 by June 30, 2018. As of
January 2018, [StackExchange] quoted 4 percent of browsers using
TLSv1.0.

The number of websites supporting TLS 1.2 is still growing (+0.4%),
and has reached 92% according to sslpulse as of June 19, 2018.
[SSLpulse] Deprecating TLS 1.0 and TLS 1.1 will thus not have a major
impact on browser or web server implementations.

Figure 1 presents statistics for use of TLS versions in the web.

```
+----------------+----------+------+-------+-------+-------+-------+
| Name/Ref       | Date     | SSLv3|TLSv1.0|TLSv1.1|TLSv1.2|TLSv1.3|
+----------------+----------+------+-------+-------+-------+-------+
! Alexa [1]      | 20180226 |   -  |  2.0  |  <0.1 |  97.9 |   -   |
| Cloudflare [2] | 20180518 |  0.0 |  9.3  |   0.2 |  84.9 |   5.5 |
| Firefox [3]    | 20180709 |   -  |  1.0  |    -  |  94.0 |   5.0 |
| Chrome [4]     | 20180711 |   -  |  0.4  |  <0.1 |   -   |   -   |
+----------------+----------+------+-------+-------+-------+-------+
```
[1] https://scotthelme.co.uk/alexa-top-1-million-analysis-february-2018/
[2] https://www.ietf.org/mail-archive/web/tls/current/msg26578.html
[3] https://www.ietf.org/mail-archive/web/tls/current/msg26575.html
[4] https://www.ietf.org/mail-archive/web/tls/current/msg26620.html

Figure 1: Web Statistics

4.2. Mail

E-Mail uses TLS for SMTP, submission (port 587), POP/POP3 and IMAP.
Typically email deployments lag public web deployments in terms of
the rate of adoption of new TLS versions. Figure 2 presents
statistics for use of TLS versions in the email applications.

```
+----------------+----------+------+-------+-------+-------+-------+
| Name/Ref       | Date     | SSLv3|TLSv1.0|TLSv1.1|TLSv1.2|TLSv1.3|
+----------------+----------+------+-------+-------+-------+-------+
| Clusters [1]   | 20180316 | <0.1 | 10.6  |  <0.1 |  89.3 |   -   |
| TLSA [2]       | 20180710 |   -  |  1.4  |   0.1 |  98.5 |   -   |
| UK-ESP [3]     | 20180710 |   -  | 19.9  |  <0.1 |   -   |   -   |
+----------------+----------+------+-------+-------+-------+-------+
```
[1] https://eprint.iacr.org/2018/299
[2] https://www.ietf.org/mail-archive/web/tls/current/msg26603.html
[3] https://www.ietf.org/mail-archive/web/tls/current/msg26603.html

Figure 2: Mail Statistics

4.3. Operating Systems

Figure 3 presents statistics for use of TLS versions in operating
systems.

```
+----------------+----------+------+-------+-------+-------+-------+
| Name/Ref       | Date     | SSLv3|TLSv1.0|TLSv1.1|TLSv1.2|TLSv1.3|
+----------------+----------+------+-------+-------+-------+-------+
| Windows cli [1]| 20180709 |   -  | >10.0 |  ~0.3 |   -   |   -   |
| Windows svr [1]| 20180709 |   -  |  ~1.5 |  ~0.0 |   -   |   -   |
| Apple [2]      | 20180709 |   -  |   0.4 |    -  |  99.6 |   -   |
+----------------+----------+------+-------+-------+-------+-------+
```
[1] https://www.ietf.org/mail-archive/web/tls/current/msg26577.html
[2] https://www.ietf.org/mail-archive/web/tls/current/msg26634.html

Figure 3: Operating System Statistics

4.4. Enterprise Networks

Figure 4 presents statistics for use of TLS versions in the
enterprise networks. The tcd.ie numbers below were the result of a
student project and need further validation.

```
+----------------+----------+------+-------+-------+-------+-------+
| Name/Ref       | Date     | SSLv3|TLSv1.0|TLSv1.1|TLSv1.2|TLSv1.3|
+----------------+----------+------+-------+-------+-------+-------+
| tcd.ie [1]     | 20180713 | 18.0 |  35.0 |   0   |  45.0 |    0  |
+----------------+----------+------+-------+-------+-------+-------+
```
[1] https://www.ietf.org/mail-archive/web/tls/current/msg26633.html

Figure 4: Enterprise Network Statistics

5. 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.

6. 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 then 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.

[[Text here is derived (or stolen:-) from [RFC7568]]]

7. 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 then 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.

8. Do Not Use SHA-1 in TLSv1.2

   [[This section was suggested in PR#2 for the pre-WG draft repo by
   Hubert Kario. We're not clear if the WG would like this draft to
   include this or not, so will ask the TLS WG at the appropriate
   time.]]

   SHA-1 as a signature hash MUST NOT be used. That means that clients
   MUST send signature_algorithms extension and that extension MUST NOT
   include pairs that include SHA-1 hash. In particular, values {2, 1},
   {2, 2} and {2, 3} MUST NOT be present in the extension.

   Note: this does not affect cipher suites that use SHA-1 HMAC for data
   integrity as the HMAC construction is still considered secure and
   when they are used in TLSv1.2 SHA-256 is used for handshake
   integrity.

9. Updates to RFC7525

   [[Since RFC7525 is BCP195, there'll probably be some process-fun to
   do an update of that. Formally, it may be that this document becomes
   a new part of BCP195 I guess, but we can figure that out with chairs
   and ADs.]]

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

   o 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.

   o 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:

o Implementations MUST NOT negotiate DTLS version 1.0 [RFC4347].

   Version 1.0 of DTLS correlates to version 1.1 of TLS (see above).

10. 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.

11. 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, and Jakub Wilk.

12. IANA Considerations

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

13. References

13.1. 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,
           <https://www.rfc-editor.org/info/rfc2119>.

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0",
           RFC 2246, DOI 10.17487/RFC2246, January 1999,
           <https://www.rfc-editor.org/info/rfc2246>.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security
           (TLS) Protocol Version 1.1", RFC 4346,
           DOI 10.17487/RFC4346, April 2006,
           <https://www.rfc-editor.org/info/rfc4346>.

[RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre,
           "Recommendations for Secure Use of Transport Layer
           Security (TLS) and Datagram Transport Layer Security
           (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May
           2015, <https://www.rfc-editor.org/info/rfc7525>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
              2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
              May 2017, <https://www.rfc-editor.org/info/rfc8174>.

13.2. Informative References

   [Amazon]   Amazon, "Amazon Elastic Load Balancing Support Deprecated
              TLSv1.0 and TLSv1.1 https://aws.amazon.com/about-aws/
              whats-new/2017/02/elastic-load-balancing-support-for-tls-
              1-1-and-tls-1-2-pre-defined-security-policies/", 2017.

   [Bhargavan2016]
              Bhargavan, K. and G. Leuren, "Transcript Collision
              Attacks: Breaking Authentication in TLS, IKE, and SSH
              https://www.mitls.org/downloads/
              transcript-collisions.pdf", 2016.

   [Canada]   Treasury Board of Canada Secretariat, "Implementing HTTPS
              for Secure Web Connections: Information Technology Policy
              Implementation Notice (ITPIN)", June 2018,
              <https://www.canada.ca/en/treasury-board-
              secretariat/services/information-technology/
              policy-implementation-notices/
              implementing-https-secure-web-connections-itpin.html>.

   [CloudFlare]
              CloudFlare, "CloudFlare Deprecated TLSv1.0 and TLSv1.1
              https://blog.cloudflare.com/deprecating-old-tls-versions-
              on-cloudflare-dashboard-and-api/", 2018.

   [Digicert]
              Digicert, "Deprecating TLS 1.0 and 1.1
              https://www.digicert.com/blog/
              depreciating-tls-1-0-and-1-1/", 2018.

   [GIT]   GitHub, "GitHub Deprecates TLSv1.0 and TLSv1.1
              https://githubengineering.com/crypto-removal-notice/",
              2018.

   [KeyCDN]   KeyCDN, "Deprecating TLS 1.0 and 1.1 Enhancing Security
              for Everyone
              https://www.keycdn.com/blog/deprecating-tls-1-0-and-1-1/",
              2018.

[NIST800-52r2]
          National Institute of Standards and Technology, "NIST
          SP800-52r2 https://csrc.nist.gov/CSRC/media/Publications/
          sp/800-52/rev-2/draft/documents/sp800-52r2-draft.pdf",
          2018.

[paypal] Paypal, ""TLS1.2 and HTTP/1.1 Upgrade" https://www.paypal-
          notice.com/en/TLS-1.2-and-HTTP1.1-Upgrade/", 2018.

[PCI-TLS1]
          PCI Security Standards Council, "Migrating from SSL and
          Early TLS https://www.pcisecuritystandards.org/documents/
          Migrating-from-SSL-Early-TLS-Info-Supp-v1_1.pdf", 2016.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer
          Security", RFC 4347, DOI 10.17487/RFC4347, April 2006,
          <https://www.rfc-editor.org/info/rfc4347>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security
          (TLS) Protocol Version 1.2", RFC 5246,
          DOI 10.17487/RFC5246, August 2008,
          <https://www.rfc-editor.org/info/rfc5246>.

[RFC7457] Sheffer, Y., Holz, R., and P. Saint-Andre, "Summarizing
          Known Attacks on Transport Layer Security (TLS) and
          Datagram TLS (DTLS)", RFC 7457, DOI 10.17487/RFC7457,
          February 2015, <https://www.rfc-editor.org/info/rfc7457>.

[RFC7568] Barnes, R., Thomson, M., Pironti, A., and A. Langley,
          "Deprecating Secure Sockets Layer Version 3.0", RFC 7568,
          DOI 10.17487/RFC7568, June 2015,
          <https://www.rfc-editor.org/info/rfc7568>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol
          Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018,
          <https://www.rfc-editor.org/info/rfc8446>.

[RFC8447] Salowey, J. and S. Turner, "IANA Registry Updates for TLS
          and DTLS", RFC 8447, DOI 10.17487/RFC8447, August 2018,
          <https://www.rfc-editor.org/info/rfc8447>.

[SSLpulse]
          SSLpulse - will be deleted before publication, "SSLpulse
          https://www.ssllabs.com/ssl-pulse/", 2018.

   [StackExchange]
             StackExchange - will be deleted before publication,
             "Stackexchange
             https://security.stackexchange.com/questions/177182/is-
             there-a-list-of-old-browsers-that-only-support-tls-1-0",
             2018.

   [stripe]   Stripe, ""Upgrading to SHA-2 and TLS 1.2"
             https://stripe.com/blog/upgrading-tls", 2018.

Appendix A. Change Log

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

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

o I-Ds became RFCs 8446/8447 (old-repo PR#4, for TLS1.3)
o Accepted old-repo PR#5 fixing typos

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

o Added stats sent to list so far
o PR's #2,3
o a few more references
o 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