                 Prohibiting Secure Sockets Layer (SSL) Version 2.0

Abstract

   This document requires that when Transport Layer Security (TLS)
   clients and servers establish connections, they never negotiate the
   use of Secure Sockets Layer (SSL) version 2.0. This document updates
   the backward compatibility sections found in the Transport Layer
   Security (TLS).

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc6176.

1. Introduction

   Many protocols specified in the IETF rely on Transport Layer Security
   (TLS) [TLS1.0][TLS1.1][TLS1.2] for security services. This is a good
   thing, but some TLS clients and servers also support negotiating the
   use of Secure Sockets Layer (SSL) version 2.0 [SSL2]; however, this
   version does not provide a sufficiently high level of security. SSL
   version 2.0 has known deficiencies. This document describes those
   deficiencies, and it requires that TLS clients and servers never
   negotiate the use of SSL version 2.0.

   RFC 4346 [TLS1.1], and later RFC 5246 [TLS1.2], explicitly warned
   implementers that the "ability to send version 2.0 CLIENT-HELLO
   messages will be phased out with all due haste". This document
   accomplishes this by updating the backward compatibility sections
   found in TLS [TLS1.0][TLS1.1][TLS1.2].

1.1. Requirements 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
   [RFC2119].

2. SSL 2.0 Deficiencies

   SSL version 2.0 [SSL2] deficiencies include the following:

   o Message authentication uses MD5 [MD5]. Most security-aware users
      have already moved away from any use of MD5 [RFC6151].

   o Handshake messages are not protected. This permits a man-in-the-
      middle to trick the client into picking a weaker cipher suite than
      it would normally choose.

   o Message integrity and message encryption use the same key, which
      is a problem if the client and server negotiate a weak encryption
      algorithm.

   o Sessions can be easily terminated. A man-in-the-middle can easily
      insert a TCP FIN to close the session, and the peer is unable to
      determine whether or not it was a legitimate end of the session.

3. Changes to TLS

    Because of the deficiencies noted in the previous section:

    o TLS clients MUST NOT send the SSL version 2.0 compatible CLIENT-
       HELLO message format. Clients MUST NOT send any ClientHello
       message that specifies a protocol version less than
       { 0x03, 0x00 }. As previously stated by the definitions of all
       previous versions of TLS, the client SHOULD specify the highest
       protocol version it supports.

    o TLS servers MAY continue to accept ClientHello messages in the
       version 2 CLIENT-HELLO format as specified in RFC 5246 [TLS1.2],
       Appendix E.2. Note that this does not contradict the prohibition
       against actually negotiating the use of SSL 2.0.

    o TLS servers MUST NOT reply with an SSL 2.0 SERVER-HELLO with a
       protocol version that is less than { 0x03, 0x00 } and instead MUST
       abort the connection, i.e., when the highest protocol version
       offered by the client is { 0x02, 0x00 }, the TLS connection will
       be refused.

    Note that the number of servers that support this above-mentioned
    "MAY accept" implementation option is declining, and the SSL 2.0
    CLIENT-HELLO precludes the use of TLS protocol enhancements that
    require TLS extensions. TLS extensions can only be sent as part of
    an (Extended) ClientHello handshake message.

4. Security Considerations

    This entire document is about security considerations.

5. Acknowledgements

    The idea for this document was inspired by discussions between Peter
    Saint Andre, Simon Josefsson, and others on the Extensible Messaging
    and Presence Protocol (XMPP) mailing list.

    We would also like to thank Michael D'Errico, Paul Hoffman, Nikos
    Mavrogiannopoulos, Tom Petch, Yngve Pettersen, Marsh Ray, Martin Rex,
    Yaron Sheffer, and Glen Zorn for their reviews and comments.

6. References

6.1. Normative References

   [RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
                Requirement Levels", BCP 14, RFC 2119, March 1997.

   [TLS1.0]   Dierks, T. and C. Allen, "The TLS Protocol Version 1.0",
                RFC 2246, January 1999.

   [TLS1.1]   Dierks, T. and E. Rescorla, "The Transport Layer Security
                (TLS) Protocol Version 1.1", RFC 4346, April 2006.

   [TLS1.2]   Dierks, T. and E. Rescorla, "The Transport Layer Security
                (TLS) Protocol Version 1.2", RFC 5246, August 2008.

6.2. Informative References

   [MD5]   Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321,
                April 1992.

   [SSL2]   Hickman, Kipp, "The SSL Protocol", Netscape
                Communications Corp., Feb 9, 1995.

   [RFC6151]   Turner, S. and L. Chen, "Updated Security Considerations
                for the MD5 Message-Digest and the HMAC-MD5 Algorithms",
                RFC 6151, March 2011.

Authors' Addresses

   Sean Turner
   IECA, Inc.
   3057 Nutley Street, Suite 106
   Fairfax, VA 22031
   USA

   EMail: turners@ieca.com

   Tim Polk
   National Institute of Standards and Technology
   100 Bureau Drive, Mail Stop 8930
   Gaithersburg, MD 20899-8930
   USA

   EMail: tim.polk@nist.gov