Abstract

Secure Sockets Layer version 3.0 (SSLv3) [RFC6101] is not sufficiently secure. This document requires that SSLv3 not be used. The replacement versions, in particular Transport Layer Security (TLS) 1.2 [RFC5246], are considerably more secure and capable protocols.

This document updates the backward compatibility section of RFC 5246 and its predecessors to prohibit fallback to SSLv3.

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 http://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 October 12, 2015.

Copyright Notice

Copyright (c) 2015 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
1. Introduction

The SSLv3 protocol has been subject to a long series of attacks, both on its key exchange mechanism and on the encryption schemes it supports since it was released in 1996. Despite being replaced by TLS 1.0 [RFC2246] in 1999, and subsequently TLS 1.1 in 2002 [RFC4346] and 1.2 in 2006 [RFC5246], availability of these replacement versions has not been universal. As a result, many implementations of TLS have permitted the negotiation of SSLv3.

The predecessor of SSLv3, SSL version 2, is no longer considered sufficiently secure [RFC6176]. SSLv3 now follows.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].
3. Do Not Use SSL Version 3.0

SSLv3 MUST NOT be used. Negotiation of SSLv3 from any version of TLS MUST NOT be permitted.

Any version of TLS is more secure than SSLv3, though the highest version available is preferable.

Pragmatically, clients MUST NOT send a ClientHello with ClientHello.client_version set to {03,00}. Similarly, servers MUST NOT send a ServerHello with ServerHello.server_version set to {03,00}. Any party receiving a Hello message with the protocol version set to {03,00} 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 SSLv3.

4. SSLv3 is Comprehensively Broken

4.1. Record Layer

The non-deterministic padding used in the CBC construction of SSLv3 trivially permits the recovery of plaintext [POODLE]. More generally, the cipher block chaining (CBC) modes of SSLv3 use a flawed MAC-then-encrypt construction that has subsequently been replaced in TLS versions [RFC7366]. Unfortunately, the mechanism to correct this flaw relies on extensions: a feature added in TLS 1.0. SSLv3 cannot be updated to correct this flaw in the same way.

The flaws in the CBC modes in SSLv3 are mirrored by the weakness of the stream ciphers it defines. Of those defined, only RC4 is currently in widespread use. RC4, however, exhibits serious biases and is also no longer fit for use [RFC7465].

This leaves SSLv3 with no suitable record protection mechanism.

4.2. Key Exchange

The SSLv3 key exchange is vulnerable to man-in-the-middle attacks when renegotiation [Ray09] or session resumption [TRIPLE-HS] are
used. Each flaw has been fixed in TLS by means of extensions. Again, SSLv3 cannot be updated to correct these flaws.

4.3. Custom Cryptographic Primitives

SSLv3 defines custom constructions for PRF, HMAC and digital signature primitives. Such constructions lack the deep cryptographic scrutiny that standard constructions used by TLS have received. Furthermore, all SSLv3 primitives rely on SHA-1 [RFC3174] and MD5 [RFC1321]: these hash algorithms are considered weak and are being systematically replaced with stronger hash functions, such as SHA-256 [FIPS180-2].

5. Limited Capabilities

SSLv3 is unable to take advantage of the many features that have been added to recent TLS versions. This includes the features that are enabled by ClientHello extensions, which SSLv3 does not support.

Though SSLv3 can benefit from new cipher suites, it cannot benefit from new cryptographic modes and features. Of these, the following are particularly prominent:

- Authenticated Encryption with Additional Data (AEAD) modes are added in [RFC5246].
- Elliptic Curve Diffie-Hellman (ECDH) and Digital Signature Algorithm (ECDSA) are added in [RFC4492].
- Stateless session tickets [RFC5077].
- A datagram mode of operation, DTLS [RFC6347].
- Application layer protocol negotiation [RFC7301].

6. IANA Considerations

This document has no IANA actions.

7. Security Considerations

This entire document aims to improve security by prohibiting the use of a protocol that is not secure.
8. References

8.1. Normative References


8.2. Informative References


Authors’ Addresses

Richard Barnes
Mozilla

Email: rlb@ipv.sx

Martin Thomson
Mozilla

Email: martin.thomson@gmail.com

Alfredo Pironti
INRIA

Email: alfredo@pironti.eu

Adam Langley
Google

Email: agl@google.com