Internet Engineering Task Force
Internet-Draft
Updates: 5246 7525 (if approved)
Intended status: Standards Track
Expires: July 25, 2020

L. Velvindron
cyberstorm.mu
K. Moriarty
Dell EMC
A. Ghedini
Cloudflare Inc.
January 22, 2020

Deprecating MD5 and SHA-1 signature hashes in TLS 1.2
draft-ietf-tls-md5-sha1-deprecate-02

Abstract

The MD5 and SHA-1 hashing algorithms are steadily weakening in strength and their deprecation process should begin for their use in TLS 1.2 digital signatures. However, this document does not deprecate SHA-1 in HMAC for record protection.

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 July 25, 2020.

Copyright Notice

Table of Contents

1. Introduction 2
1.1. Requirements Language 2
2. Signature Algorithms 2
3. Certificate Request 3
4. Server Key Exchange 3
5. Certificate Verify 3
6. Updates to RFC5246 3
7. Updates to RFC7525 4
8. Security Considerations 4
9. Acknowledgement 4
10. References 5
10.1. Normative References 5
10.2. Informative References 5
Authors' Addresses 5

1. Introduction

The usage of MD5 and SHA-1 for signature hashing in TLS 1.2 is specified in RFC 5246 [RFC5246]. MD5 and SHA-1 have been proven to be insecure, subject to collision attacks. RFC 6151 [RFC6151] details the security considerations, including collision attacks for MD5, published in 2011. NIST formally deprecated use of SHA-1 in 2011 [NISTSP800-131A-R2] and disallowed its use for digital signatures at the end of 2013, based on both the Wang, et. al, attack and the potential for brute-force attack. Further, in 2017, researchers from Google and CWI Amsterdam [SHA-1-Collision] proved SHA-1 collision attacks were practical. This document updates RFC 5246 [RFC5246] and RFC7525 [RFC7525] in such as way that MD5 and SHA-1 MUST NOT be used for digital signatures. However, this document does not deprecate SHA-1 in HMAC for record protection.

1.1. Requirements Language

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].

2. Signature Algorithms

Clients SHOULD NOT include MD5 and SHA-1 in the signature_algorithms extension. If a client does not send a signature_algorithms extension, then the server MUST abort the handshake and send a

handshake_failure alert, except when digital signatures are not used (for example, when using PSK ciphers).

3. Certificate Request

Servers SHOULD NOT include MD5 and SHA-1 in CertificateRequest message.

4. Server Key Exchange

Servers MUST NOT include MD5 and SHA-1 in ServerKeyExchange message. If client does receive a MD5 or SHA-1 signature in the ServerKeyExchange message and it sent one in signature_algorithms extensions it MUST abort the connection with handshake_failure or insufficient_security alert. If client did not send MD5 nor SHA-1 hash algorithm in signature_algorithms extension and it receives a MD5 or SHA-1 signature in the ServerKeyExchange it MUST abort the connection with the illegal_parameter alert.

5. Certificate Verify

Clients MUST NOT include MD5 and SHA-1 in CertificateVerify message. If the server receives a CertificateVerify message with MD5 or SHA-1 it MUST abort the connection with handshake_failure or insufficient_security alert.

6. Updates to RFC5246

RFC5246 [RFC5246], The Transport Layer Security (TLS) Protocol Version 1.2, suggests that implementations can assume support for MD5 and SHA-1 by their peer. This update changes the suggestion to assume support for SHA-256 instead, due to MD5 and SHA-1 being deprecated.

OLD:

In Section 7.4.1.4.1: the text should be revised from " Note: this is a change from TLS 1.1 where there are no explicit rules, but as a practical matter one can assume that the peer supports MD5 and SHA-1."

NEW:

"Note: This is a change from TLS 1.1 where there are no explicit rules, but as a practical matter one can assume that the peer supports SHA-256."

7. Updates to RFC7525

RFC7525 [RFC7525], Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) recommends use of SHA-256 as a minimum requirement. This update moves the minimum recommendation to use stronger language deprecating use of both SHA-1 and MD5. The prior text did not explicitly include MD5 and this text adds it to ensure it is understood as having been deprecated.

Section 4.3:

OLD:

When using RSA, servers SHOULD authenticate using certificates with at least a 2048-bit modulus for the public key. In addition, the use of the SHA-256 hash algorithm is RECOMMENDED (see [CAB-Baseline] for more details). Clients SHOULD indicate to servers that they request SHA-256, by using the "Signature Algorithms" extension defined in TLS 1.2.

NEW:

Servers SHOULD authenticate using certificates with at least a 2048-bit modulus for the public key.

In addition, the use of the SHA-256 hash algorithm is RECOMMENDED, SHA-1 or MD5 MUST NOT be used (see [CAB-Baseline] for more details). Clients MUST indicate to servers that they request SHA-256, by using the "Signature Algorithms" extension defined in TLS 1.2.

8. Security Considerations

Concerns with TLS 1.2 implementations falling back to SHA-1 is an issue. This draft updates the TLS 1.2 specification to deprecate support for MD5 and SHA-1 for digital signatures. However, this document does not deprecate SHA-1 in HMAC for record protection.

9. Acknowledgement

The authors would like to thank Hubert Kario for his help in writing the initial draft. We are also grateful to Daniel Migault, Martin Thomson and David Cooper for their feedback.

10. References

10.1. 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>.

[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>.

10.2. Informative References

[CAB-Baseline] CA/Browser Forum, "Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates Version 1.1.6", 2013, <https://www.cabforum.org/documents.html>.

[NISTSP800-131A-R2] Barker, E. and A. Roginsky, "Transitioning the Use of Cryptographic Algorithms and Key Lengths", March 2019, <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf>.

[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>.

[RFC6151] Turner, S. and L. Chen, "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms", RFC 6151, DOI 10.17487/RFC6151, March 2011, <https://www.rfc-editor.org/info/rfc6151>.

[SHA-1-Collision] Stevens, M., Bursztein, E., Karpman, P., Albertini, A., and Y. Markov, "The first collision for full SHA-1", March 2019, <http://shattered.io/static/shattered.pdf>.

Authors' Addresses

Loganaden Velvindron
cyberstorm.mu
Rose Hill
MU

Phone: +230 59762817
Email: logan@cyberstorm.mu

Kathleen Moriarty
Dell EMC

Email: Kathleen.Moriarty.ietf@gmail.com

Alessandro Ghedini
Cloudflare Inc.

Email: alessandro@cloudflare.com