tls
Internet-Draft
Intended status: BCP
Expires: March 24, 2014

Y. Sheffer
Porticor
R. Holz
TUM

September 20, 2013

Recommendations for Secure Use of TLS and DTLS
draft-sheffer-tls-bcp-01

Abstract

Over the last few years there have been several serious attacks on TLS, including attacks on its most commonly used ciphers and modes of operation. This document offers recommendations on securely using the TLS and DTLS protocols, given existing standards and implementations.

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 March 24, 2014.

Copyright Notice

Table of Contents

1. Introduction

Over the last few years there have been several major attacks on TLS [RFC5246], including attacks on its most commonly used ciphers and modes of operation. Details are given in Section 2, but suffice it to say that both AES-CBC and RC4, which together make up for most current usage, have been seriously attacked in the context of TLS.

Given these issues, there is need for IETF guidance on how TLS can be used securely. Unlike most IETF documents, this is guidance for deployers, as well as for implementers. In fact the recommendations below call for the use of widely implemented algorithms, which are not seeing widespread use today.

Rather than standardizing new mechanisms in TLS, our goal is to recommend a few already-specified mechanisms and cipher suites, and to encourage the industry to use them in order to improve the overall security of TLS-protected network traffic. When picking these mechanisms, we consider their security, their technical maturity and interoperability, as well as their prevalence at the time of writing.

This recommendation applies to both TLS and DTLS. TLS 1.3, when it is standardized and deployed in the field, should resolve the current vulnerabilities while providing significantly better functionality, and will very likely obsolete the current document.

Our knowledge about the strength of various algorithms and feasible attacks can change quickly, and experience shows that a crypto BCP is a point-in-time statement more than other BCPs. Readers are advised to seek out any errata or udpates that apply to this document.

1.1. Conventions used in this document

[[Are we normative? Currently we're not and this section might go away.]]

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 [RFC2119].

2. Attacks on TLS

This section lists the attacks that motivated the current recommendations. This is not intended to be an extensive survey of TLS's security.

While there are widely deployed mitigations for some of the attacks

listed below, we believe that their root causes necessitate a more systemic solution.

2.1. BEAST

The BEAST attack [BEAST] uses issues with the TLS 1.0 implementation of CBC (that is, predictable IV) to decrypt parts of a packet, and specifically shows how this can be used to decrypt HTTP cookies when run over TLS.

2.2. Lucky Thirteen

A consequence of the MAC-then-encrypt design in all current versions of TLS is the existence of padding oracle attacks [Padding-Oracle]. A recent incarnation of these attacks is the Lucky Thirteen attack [CBC-Attack], a timing side-channel attack that allows the attacker to decrypt arbitrary ciphertext.

2.3. Attacks on RC4

The RC4 algorithm [RC4] has been used with TLS (and previously, SSL) for many years. Attacks have also been known for a long time, e.g. [RC4-Attack-FMS]. But recent attacks ([RC4-Attack], [RC4-Attack-AlF]) have weakened this algorithm even more. See [I-D.popov-tls-prohibiting-rc4] for more details.

2.4. Compression Attacks: CRIME and BREACH

The CRIME attack [CRIME] allows an active attacker to decrypt

cyphertext (specifically, cookies) when TLS is used with protocol-level compression.

The BREACH attack [BREACH] makes similar use of HAdded TTP-level compression, which is much more prevalent than compression at the TLS level, to decrypt secret data passed in the HTTP response.

The former attack can be mitigated by disabling TLS compression, as recommended below. We are not aware of mitigations at the protocol level to the latter attack, and so application-level mitigations are needed (see [BREACH]). For example, implementations of HTTP that use CSRF tokens will need to randomize them even when the recommendations of the current document are adopted.

3. Selection Criteria

Given the above attacks, we are proposing that deployers opt for a specific cipher suite when negotiating TLS. We have used the

following criteria when framing our recommendations:

- The cipher suite must be secure in default use, and should not require any additional security measures beyond those defined in the standard.
- The cipher suite must be widely implemented, i.e. available in a large percentage of popular cryptographic libraries.
- The cipher suite must have undergone a significant amount of analysis, and the algorithm and mode of operation must both be standardized by relevant organizations.
- We prefer cipher suites that provide client-side privacy and perfect forward secrecy, i.e. those that use ephemeral Diffie-Hellman. See Section 6.2 for more details.
- As currently specified and implemented, elliptic curve groups are preferable over modular DH groups: they are easier and safer to use within TLS.
- When there are multiple key sizes available, we have chosen the current industry standard, 128 bits of strength. Of course deployers are free to opt for a stronger cipher suite.

4. Recommendations

Following are recommendations for people implementing and deploying client and server-side TLS.

4.1. Summary

Based on the criteria above, we recommend using as a preferred cipher suite the following:

o TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 [RFC5829]

It is noted that the above cipher suite is an authenticated encryption (AEAD) algorithm [RFC5116], and therefore requires the use of TLS 1.2.

We recommend using 2048-bit server certificates, with a SHA-256 fingerprint. See [CAB-Baseline] for more details.

[RFC4492] allows clients and servers to negotiate ECDH parameters (curves). We recommend that clients and servers prefer verifiably random curves (specifically Brainpool P-256, brainpoolp256r1 [I-D.merkle-tls-brainpool]), and fall back to the commonly used NIST P-256 (secp256r1) [RFC4492]. In addition, clients should send an ec_point_formats extension with a single element, "uncompressed".

We recommend to always disable TLS-level compression ([RFC5246], Sec.

6.2.2).

Finally, we recommend that clients disable fallback to SSLv3 (see Section 4.3).

4.2. Cipher Suite Negotiation Details

We recommend that clients include the above cipher suite as the first proposal to any server, unless they have prior knowledge that the server cannot respond to a TLS 1.2 client_hello message.

We recommend that servers prefer this cipher suite (or a similar but stronger one) whenever it is proposed, even if it is not the first proposal.

Both clients and servers should include the "Supported Elliptic Curves" extension [RFC4492].

Clients are of course free to offer stronger cipher suites, e.g. using AES-256; when they do, the server should prefer the stronger cipher suite unless there are reasons (e.g. performance) to choose otherwise.

Note that other profiles of TLS 1.2 exist that use different cipher suites. For example, [RFC6460] defines a profile that uses the TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 and TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 cipher suites.

This document is not an application profile standard, in the sense of Sec. 9 of [RFC5246]. As a result, clients and servers are still required to support the TLS mandatory cipher suite, TLS_RSA_WITH_AES_128_CBC_SHA.

4.3. Downgrade Attacks

Some client implementations revert to SSLv3 if the server rejected higher versions of SSL/TLS. This fallback can be forced by a MITM attacker. Moreover, IP scans [[reference?]] show that SSLv3-only servers amount to about 3% of the current server population. As a result, we recommend that by default, clients should avoid falling back to SSLv3.

4.4. Alternatives

Elliptic Curves Cryptography is not universally deployed for several reasons, including its complexity compared to modular arithmetic and longstanding IPR concerns. On the other hand, there are two related issues hindering effective use of modular Diffie-Hellman cipher

suites in TLS:

- There are no protocol mechanisms to negotiate the DH groups or parameter lengths supported by client and server.
- There are widely deployed client implementations that reject received DH parameters, if they are longer than 1024 bits.

We note that with DHE and ECDHE cipher suites, the TLS master key

only depends on the Diffie Hellman parameters and not on the strength the the RSA certificate; moreover, 1024 bits DH parameters are generally considered insufficient at this time.

Because of the above, we recommend using (in priority order):

1. Elliptic Curve DHE with negotiated parameters, as described in Section 4.1.
2. TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 [RFC5288], with 2048-bit Diffie-Hellman parameters.
3. The same cipher suite, with 1024-bit parameters.

With modular ephemeral DH, deployers should carefully evaluate interoperability vs. security considerations when configuring their TLS endpoints.

5. Implementation Status

Since this document does not propose a new protocol or a new cipher suite, we do not provide a full implementation status, as per [RFC6982]. However it is useful to list some known existing implementations of the recommended cipher suite(s).

Category	Software	As Of Version	Comment
Library	OpenSSL	1.0.1	
	GnuTLS		
	NSS	3.11.1	
Browser	Internet Explorer	IE8 on Windows 7	
	Firefox	TBD	
	Chrome	TLS 1.2 and AES-GCM expected in Chrome 30	
	Safari	TBD	
Web server	Apache (mod_gnutls)	??	
	Apache	??	

	(mod_ssl) Nginx	1.0.9, 1.1.6	With a recent version of OpenSSL	

6. Security Considerations

6.1. AES-GCM

Please refer to [RFC5246], Sec. 11 for general security considerations when using TLS 1.2, and to [RFC5288], Sec. 6 for security considerations that apply specifically to AES-GCM when used with TLS.

6.2. Perfect Forward Secrecy (PFS)

PFS is a defense against an attacker who records encrypted conversations where the session keys are only encrypted with the communicating parties' long-term keys. Should the attacker be able to obtain these long-term keys at some point later in the future, he will be able to decrypt the session keys and thus the entire conversation. In the context of TLS and DTLS, such compromise of long-term keys is not entirely implausible. It can happen, for example, due to:

- A client or server being attacked by some other attack vector, and the private key retrieved.
- A long-term key retrieved from a device that has been sold or otherwise decommissioned without prior wiping.
- A long-term key used on a device as a default key [Heninger2012].
- A key generated by a Trusted Third Party like a CA, and later retrieved from it either by extortion or compromise [Soghoian2011].
- A cryptographic break-through, or the use of asymmetric keys with insufficient length [Kleinjung2010].

PFS ensures in such cases that the session keys cannot be determined even by an attacker who obtains the long-term keys some time after the conversation. It also protects against an attacker who is in possession of the long-term keys, but remains passive during the conversation.

PFS is generally achieved by using the Diffie-Hellman scheme to derive session keys. The Diffie-Hellman scheme has both parties maintain private secrets and send parameters over the network as

modular powers over certain cyclic groups. The properties of the so-called Discrete Logarithm Problem (DLP) allow to derive the session keys without an eavesdropper being able to do so. There is currently no known attack against DLP if sufficiently large parameters are chosen.

Unfortunately, many TLS/DTLS cipher suites were defined that do not enable PFS, e.g. TLS_RSA_WITH_AES_256_CBC_SHA256. We thus advocate strict use of PFS-only ciphers. These are listed in Section Section 4.1.

6.3. Session Resumption

TBD, https://www.imperialviolet.org/2013/06/27/botchingpfs.html.

7. IANA Considerations

[Note to RFC Editor: please remove this section before publication.]

This document requires no IANA actions.

8. Acknowledgements

We would like to thank Stephen Farrell, Simon Josefsson, Yoav Nir, Kenny Paterson, Patrick Pelletier, and Rich Salz for their review. Thanks to Brian Smith whose "browser cipher suites" page is a great resource. Finally, Thanks to all others who commented on the TLS and other lists and are not mentioned here by name.

The document was prepared using the lyx2rfc tool, created by Nico Williams.

9. References

9.1. Normative References

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

[RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", RFC 4492, May 2006.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5288] Salowey, J., Choudhury, A., and D. McGrew, "AES Galois Counter Mode (GCM) Cipher Suites for TLS", RFC 5288, August 2008.

[RFC5829] Brown, A., Clemm, G., and J. Reschke, "Link Relation Types for Simple Version Navigation between Web Resources", RFC 5829, April 2010.

[I-D.merkle-tls-brainpool] Merkle, J. and M. Lochter, "ECC Brainpool Curves for Transport Layer Security (TLS)", draft-merkle-tls-brainpool-04 (work in progress), July 2013.

9.2. Informative References

[I-D.popov-tls-prohibiting-rc4] Popov, A., "Prohibiting RC4 Cipher Suites", draft-popov-tls-prohibiting-rc4-00 (work in progress), August 2013.

[RFC5116] McGrew, D., "An Interface and Algorithms for Authenticated Encryption", RFC 5116, January 2008.

[RFC6460] Salter, M. and R. Housley, "Suite B Profile for Transport Layer Security (TLS)", RFC 6460, January 2012.

[RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", RFC 6982, July 2013.

[CBC-Attack] AlFardan, N. and K. Paterson, "Lucky Thirteen: Breaking the TLS and DTLS Record Protocols", IEEE Symposium on Security and Privacy , 2013.

[BEAST] Rizzo, J. and T. Duong, "Browser Exploit Against SSL/TLS", 2011, <http://packetstormsecurity.com/files/105499/Browser-Exploit-Against-SSL-TLS.html>.

[CRIME] Rizzo, J. and T. Duong, "The CRIME Attack", EKOparty Security Conference 2012, 2012.

[BREACH] Prado, A., Harris, N., and Y. Gluck, "The BREACH Attack", 2013, <http://breachattack.com/>.

[RC4] Schneier, B., "Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd Ed.", 1996.

[RC4-Attack-FMS]
Fluhrer, S., Mantin, I., and A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4", Selected Areas in Cryptography , 2001.

[RC4-Attack]
ISOBE, T., OHIGASHI, T., WATANABE, Y., and M. MORII, "Full Plaintext Recovery Attack on Broadcast RC4", International Workshop on Fast Software Encryption , 2013.

[RC4-Attack-AlF]
AlFardan, N., Bernstein, D., Paterson, K., Poettering, B., and J. Schuldt, "On the Security of RC4 in TLS", Usenix Security Symposium 2013, 2013, <https://www.usenix.org/conference/usenixsecurity13/security-rc4-tls>.

[Padding-Oracle]
Vaudenay, S., "Security Flaws Induced by CBC Padding Applications to SSL, IPSEC, WTLS...", EUROCRYPT 2002, 2002, <http://www.iacr.org/cryptodb/archive/2002/EUROCRYPT/2850/2850.pdf>.

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

[TLS-IANA]
"Transport Layer Security (TLS) Parameters - TLS Cipher Suite Registry", <https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml#tls-parameters-4>.

[Heninger2012]
Heninger, N., Durumeric, Z., Wustrow, E., and J. Halderman, "Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices", Usenix Security Symposium 2012, 2012.

[Kleinjung2010]
Kleinjung, T., "Factorization of a 768-Bit RSA Modulus", CRYPTO 10, 2010.

[Soghoian2011]
Soghoian, C. and S. Stamm, "Certified lies: Detecting and defeating government interception attacks against SSL.", Proc. 15th Int. Conf. Financial Cryptography and Data Security , 2011.

Appendix A. Appendix: Change Log

Note to RFC Editor: please remove this section before publication.

A.1. -01

- Clarified our motivation in the introduction.
- Added a section justifying the need for PFS.
- Added recommendations for RSA and DH parameter lengths. Moved from DHE to ECDHE, with a discussion on whether/when DHE is appropriate.
- Recommendation to avoid fallback to SSLv3.
- Initial information about browser support - more still needed!
- More clarity on compression.
- Client can offer stronger cipher suites.
- Discussion of the regular TLS mandatory cipher suite.

A.2. -00

- Initial version.

Authors' Addresses

Yaron Sheffer

Porticor
29 HaHarash St.
Hod HaSharon 4501303
Israel

Email: yaronf.ietf@gmail.com

Ralph Holz
Technische Universitaet Muenchen
Boltzmannstr. 3
Garching 85748
Germany

Email: holz@net.in.tum.de