

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 30, 2015

J. Mattsson
Ericsson
October 27, 2014

Overview and Analysis of Overhead Caused by TLS
draft-mattsson-uta-tls-overhead-01

Abstract

A common argument against the use of TLS is that it adds overhead. In this document we illustrate in detail how much (or little) processing, latency, and traffic overhead TLS adds. Transition to more secure cipher suites (TLS 1.2 with AES-GCM or ChaCha20-Poly1305) actually reduces both traffic and processing overhead. AES-GCM combines security, low traffic overhead, and great performance on modern hardware. On platforms without hardware support for AES-GCM, ChaCha20-Poly1305 gives the same benefits. For everything but very short connections, TLS is not inducing any major traffic overhead (nor CPU or memory overhead).

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 April 30, 2015.

Copyright Notice

Copyright (c) 2014 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 (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. TLS Handshake	2
2.1. Latency Overhead	3
3. TLS Record Layer	3
3.1. Ciphers in Use	3
3.2. Traffic Overhead	4
3.3. Processing Overhead	6
3.3.1. Modern x86 Processors	7
3.3.2. Software	7
4. Conclusions	7
5. Security Considerations	8
6. Acknowledgements	8
7. References	8
Author's Address	9

1. Introduction

The overhead from TLS can be divided into several different aspects:

- o Traffic overhead from TLS handshake
- o Latency overhead from TLS handshake
- o Traffic overhead from TLS record layer
- o Processing overhead from TLS handshake
- o Processing overhead from TLS record layer

But in many scenarios, TLS does not add much overhead at all, and moving to more secure cipher suites actually reduces both traffic and processing overhead.

2. TLS Handshake

The TLS handshake typically adds 4-7 kB of traffic overhead. TLS compression reduces traffic overhead, but has negative security implications and should be turned off [I-D.ietf-uta-tls-bcp].

Looking at the certificates, a move from 1024 to 2048 bit RSA keys increases traffic and processing overhead but is needed for security reasons. Certificates with 1024 bit RSA keys should be phased out as they only give 80 bit security. NIST recommendation is to stop using algorithms giving 80 bit security no later than 2010 [KeyLength]. A move from SHA-1 to SHA-256 adds processing overhead but is needed for security reasons. SHA-1 should not be used anymore for digital signatures (e.g. in certificates) as it gives less than 80 bit security. To summarize, SHA-2 certificates with at least 2048 bit RSA keys should be used.

2.1. Latency Overhead

In TLS 1.2 [RFC5246] and earlier versions, the initial handshake takes 2 round-trips and session resumption takes 1 round-trip. In TLS 1.3 [I-D.ietf-tls-tls13] the target is 1 round-trip for the initial round-trip and 0 round-trips for session resumption. Because of the emphasis on reducing latency (instead of only security), TLS 1.3 is expected to have much faster deployment than earlier versions.

3. TLS Record Layer

Some of the most commonly used ciphersuites have security weaknesses. Encryption algorithms such as RC4 and the CBC modes (e.g. AES and 3DES_EDE) have security weaknesses, and the hash functions SHA-1 and MD5 (but not the HMAC constructions used in TLS record layer) also have security weaknesses.

More recent ciphersuites using AES-GCM and CHACHA20_POLY1305 have no known security weaknesses, but AES-GCM, CHACHA20_POLY1305 and other AEAD suites require TLS 1.2 [RFC5246]. CHACHA20_POLY1305 is currently only an Internet draft but is still used in practice as it is very fast in software [I-D.agl-tls-chacha20poly1305]. AES-GCM is the current IETF recommendation (Internet Draft) as part of TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 [I-D.ietf-uta-tls-bcp].

3.1. Ciphers in Use

This data is included as to motivate which algorithms to cover as well as showing that new secure ciphersuites are gaining significant usage. In data from July 2014 [ICSI], AES-CBC, RC4, and HMAC-SHA1 dominates, these ciphersuites all have security weaknesses. The NULL cipher does not provide any confidentiality at all. The more secure options AES-GCM and ChaCha20-Poly1305 are starting to show significant usage. 3DES_EDE_CBC_SHA is included as it is mandatory to implement in TLS 1.0.

Algorithm	Usage
AES_128_CBC_SHA	29.1 %
RC4_128_SHA	17.4 %
AES_128_GCM	14.7 %
AES_256_CBC_SHA	14.0 %
NULL_SHA	9.8 %
RC4_128_MD5	8.3 %
CHACHA20_POLY1305	1.4 %
3DES_EDE_CBC_SHA	< 5.6 %

Table 1: Ciphers in Use (ICSI, July 2014)

3.2. Traffic Overhead

The traffic overhead comes in different forms: the TLS record layer header, Explicit IV/Nonce, MAC tag, and encryption algorithm padding. Figure 1 illustrates the packet format for a TLS protected package where [] indicates fields where usage depends on the TLS version and the ciphersuite used.

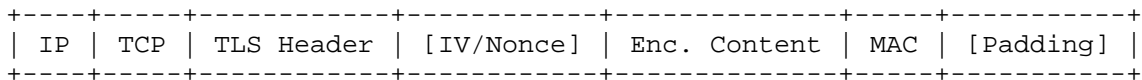


Figure 1: Format of TLS protected packet

The size of the TLS header is fixed (5 bytes). The size of the IV/Nonce depends on the TLS version and the ciphersuite used. Explicit IV is used by CBC ciphersuites in TLS 1.1 and TLS 1.2, but not TLS 1.0. Explicit Nonce is used by AEAD algorithms. The size of the MAC tag depends on the ciphersuite used, it is either a separate field (non-AEAD algorithms like SHA-1, MD5) or included in the ciphertext (AEAD algorithms like GCM, Poly1305). Padding is used by CBC ciphersuites.

The per-packet overhead for the most important ciphersuites are shown below (the values are all theoretical and the averages are calculated over a uniform distribution). For comparison, the TCP/IP overhead for IPv4 and IPv6 are 52 and 72 bytes, respectively.

AES_128_CBC_SHA, AES_256_CBC_SHA		

Per-packet overhead (TLS 1.0)	26-41 bytes (avg. 33.5)	
TLS header	5 bytes	
HMAC-SHA-1	20 bytes	
CBC padding	1-16 bytes	
Per-packet overhead (TLS 1.1, 1.2)	42-57 bytes (avg. 49.5)	
TLS header	5 bytes	
Explicit IV	16 bytes	
HMAC-SHA-1	20 bytes	
CBC padding	1-16 bytes	

3DES_EDE_CBC_SHA		

Per-packet overhead (TLS 1.0)	26-33 bytes (avg. 29.5)	
TLS header	5 bytes	
HMAC-SHA-1	20 bytes	
CBC padding	1-8 bytes	
Per-packet overhead (TLS 1.1, 1.2)	34-41 bytes (avg. 37.5)	
TLS header	5 bytes	
Explicit IV	8 bytes	
HMAC-SHA-1	20 bytes	
CBC padding	1-8 bytes	

RC4_128_SHA, NULL_SHA		

Per-packet overhead (TLS 1.0, 1.1, 1.2)	25 bytes	
TLS header	5 bytes	
HMAC-SHA-1	20 bytes	

RC4_128_MD5	

Per-packet overhead (TLS 1.0, 1.1, 1.2)	21 bytes
TLS header	5 bytes
HMAC-MD5	16 bytes

AES_128_GCM, AES_256_GCM	

Per-packet overhead (TLS 1.0, 1.1, 1.2)	29 bytes
TLS header	5 bytes
Explicit Nonce	8 bytes
GMAC	16 bytes

CHACHA20_POLY1305	

Per-packet overhead (TLS 1.0, 1.1, 1.2)	29 bytes
TLS header	5 bytes
Explicit Nonce	8 bytes
Poly1305	16 bytes

As can be seen from the tables above, there is a correlation between better security and low traffic overhead. Going from TLS 1.1 [RFC4346] with AES_CBC_SHA (mandatory to implement in TLS 1.1) to TLS 1.2 [RFC5246] with one of the more secure options AES_GCM (current IETF recommendation) or CHACHA20_POLY1305 reduces record layer traffic overhead with 41 %. Going from TLS 1.0 [RFC2246] with AES_CBC_SHA to TLS 1.2 with AES_GCM or CHACHA20_POLY1305 reduces record layer traffic overhead with 14 %.

3.3. Processing Overhead

Just as with traffic overhead, there is a correlation between better security and low processing overhead. Going from AES_CBC_SHA (mandatory to implement in TLS 1.1. and 1.2) to the more secure option AES-GCM reduces processing overhead on a Core-i7-3770 processor with 57 %. Another fact is that the overhead for AES_128_GCM and CHACHA20_POLY1305 is so low, there is no overhead reasons to not use encryption (i.e. NULL_SHA).

3.3.1. Modern x86 Processors

On modern x86 processors with hardware support for AES (AES-NI) and carry-less multiplication (CLMUL), AES_GCM is much faster than RC4_SHA, AES_CBC_SHA, or CHACHA20_POLY1305. Another performance advantage with AES-GCM is that it is designed for parallelization.

Algorithm	Speed
AES_128_GCM	1909.1 MB/s
CHACHA20_POLY1305	625.2 MB/s
AES_128_CBC_SHA	573.7 MB/s
AES_256_CBC_SHA	486.6 MB/s
RC4_128_MD5	233.9 MB/s

Table 2: Speed on 2 GHz Intel Core i7

These measurements are not fair to ChaCha20-Poly1305, but this does not matter, the important thing is how fast the algorithms run on current hardware.

3.3.2. Software

Without hardware support for AES-GCM, ChaCha20-Poly1305 is much faster than AES-GCM (and AES-CBC). Data from [Software].

Algorithm	Speed
CHACHA20_POLY1305	130.9 MB/s
AES_128_GCM	41.5 MB/s

Table 3: Speed on Snapdragon S4 Pro

Several companies have deployed ChaCha20-Poly1305 to get better performance (and security) on platforms without AES and CLMUL hardware support. This may have less significance in the future if mobile CPUs implement hardware support for AES-GCM.

4. Conclusions

Transition to more secure cipher suites (TLS 1.2 with AES-GCM or ChaCha20-Poly1305) actually reduces both traffic and processing overhead. Going from TLS 1.1 with AES_CBC_SHA (mandatory to implement in TLS 1.1) to TLS 1.2 with AES_GCM (current IETF recommendation) or CHACHA20_POLY1305 reduces record layer traffic overhead with 41 %, and record layer processing overhead with even

more. AES-GCM combines security, low traffic overhead, and great performance on modern x86 hardware. On platforms without hardware support for AES-GCM, ChaCha20-Poly1305 gives the same benefits.

Looking at the certificates, a transition to SHA-2 certificates with RSA-2048 keys increases TLS handshake traffic and processing overhead but is needed for security reasons.

For everything but very short connections, TLS is not inducing any major traffic overhead (nor CPU or memory overhead). Server people from Google Gmail has stated that "TLS accounts for less than 1% of the CPU load, less than 10 KB of memory per connection and less than 2% of network overhead". Main impact of TLS is increased latency, this can be reduced by using session resumption, cache information closer to end users, or waiting for TLS 1.3.

5. Security Considerations

The whole document is about increasing the use of TLS and secure ciphersuites by showing that TLS in many cases does not add much overhead, and that there for many types of overhead is a correlation between better security and low overhead.

6. Acknowledgements

The authors would like to thank Stephen Farrell and Ivan Ristic for their valuable comments and feedback.

7. References

- [Gueron] Shay Gueron, "AES-GCM for Efficient Authenticated Encryption - Ending the Reign of HMAC-SHA-1?", <<https://crypto.stanford.edu/RealWorldCrypto/slides/gueron.pdf>>.
- [I-D.agl-tls-chacha20poly1305] Langley, A. and W. Chang, "ChaCha20 and Poly1305 based Cipher Suites for TLS", draft-agl-tls-chacha20poly1305-04 (work in progress), November 2013.
- [I-D.ietf-tls-tls13] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3", draft-ietf-tls-tls13-02 (work in progress), July 2014.

- [I-D.ietf-uta-tls-bcp]
Sheffer, Y., Holz, R., and P. Saint-Andre,
"Recommendations for Secure Use of TLS and DTLS", draft-
ietf-uta-tls-bcp-06 (work in progress), October 2014.
- [ICSI] ICSI, "The ICSI Certificate Notary",
<<http://notary.icsi.berkeley.edu/#statistics>>.
- [KeyLength] BlueKrypt, "Cryptographic Key Length Recommendation",
<<http://www.keylength.com/>>.
- [RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0",
RFC 2246, January 1999.
- [RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security
(TLS) Protocol Version 1.1", RFC 4346, April 2006.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security
(TLS) Protocol Version 1.2", RFC 5246, August 2008.
- [Software] ImperialViolet, "TLS Symmetric Crypto",
<[https://www.imperialviolet.org/2014/02/27/
tlssymmetriccrypto.html](https://www.imperialviolet.org/2014/02/27/tlssymmetriccrypto.html)>.

Author's Address

John Mattsson
Ericsson AB
SE-164 80 Stockholm
Sweden

Phone: +46 10 71 43 501
Email: john.mattsson@ericsson.com

Network Working Group
Internet-Draft
Updates: 2595, 3207 (if approved)
Intended status: Standards Track
Expires: January 2, 2015

A. Melnikov
Isode Ltd
July 1, 2014

Updated TLS Server Identity Check Procedure for Email Related Protocols
draft-melnikov-email-tls-certs-02

Abstract

This document describes TLS server identity verification procedure for SMTP Submission, IMAP, POP and ManageSieve clients. It replaces Section 2.4 of RFC 2595.

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 January 2, 2015.

Copyright Notice

Copyright (c) 2014 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 (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions Used in This Document	2
3. Email Server Certificate Verification Rules	2
4. Examples	3
5. IANA Considerations	4
6. Security Considerations	4
7. References	4
7.1. Normative References	4
7.2. Informative References	5
Appendix A. Acknowledgements	6

1. Introduction

This document describes the updated TLS server identity verification procedure for SMTP Submission [RFC4409] [RFC3207], IMAP [RFC3501], POP [RFC1939] and ManageSieve [RFC5804] clients. It replaces Section 2.4 of RFC 2595.

Note that this document doesn't apply to use of TLS in MTA-to-MTA SMTP.

The main goal of the document is to provide consistent TLS server identity verification procedure across multiple email related protocols. This should make it easier for Certificate Authorities and ISPs to deploy TLS for email use, and would enable email client developers to write more secure code.

2. Conventions Used in This Document

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].

3. Email Server Certificate Verification Rules

During a TLS negotiation, an email client (i.e., an SMTP, IMAP, POP3 or ManageSieve client) MUST check its understanding of the server hostname against the server's identity as presented in the server Certificate message, in order to prevent man-in-the-middle attacks. Matching is performed according to the rules specified in Section 6 of [RFC6125], including "certificate pinning" and the procedure on failure to match. The following inputs are used by the verification procedure used in [RFC6125]:

1. The client MUST use the server hostname it used to open the connection as the value to compare against the server name as

expressed in the server certificate (the reference identity). The client MUST NOT use any form of the server hostname derived from an insecure remote source (e.g., insecure DNS lookup). CNAME canonicalization is not done.

The rules and guidelines defined in [RFC6125] apply to an email server certificates, with the following supplemental rules:

1. Support for the DNS-ID identifier type (subjectAltName of dNSName type [RFC5280]) is REQUIRED in Email client software implementations. Certification authorities that issue Email-specific certificates MUST support the DNS-ID identifier type. Service providers SHOULD include the DNS-ID identifier type in Certificate Signing Requests.
2. Support for the SRV-ID identifier type (subjectAltName of SRVName type [RFC4985]) is REQUIRED for email client software implementations. Certification authorities that issue email-specific certificates MUST support the SRV-ID identifier type. Service providers SHOULD include the SRV-ID identifier type in Certificate Signing Requests. List of SRV-ID types for email services is specified in [RFC6186]. For ManageSieve the value "sieve" is used.
3. URI-ID identifier type (subjectAltName of uniformResourceIdentifier type [RFC5280]) MUST NOT be used by clients for server verification.
4. For backward compatibility with deployed software CN-ID identifier type (CN attribute from the subject name, see [RFC6125]) MAY be used for server identity verification.
5. Email protocols allow use of certain wilcards in identifiers presented by email servers. The "*" wildcard character MAY be used as the left-most name component of DNS-ID or CN-ID in the certificate. For example, a DNS-ID of *.example.com would match a.example.com, foo.example.com, etc. but would not match example.com. Note that the wildcard character MUST NOT be used as a fragment of the left-most name component (e.g., *oo.example.com, f*o.example.com, or foo*.example.com).

4. Examples

Consider an IMAP-accessible email server which supports both IMAP and IMAPS (IMAP-over-TLS) at the host "mail.example.net" servicing email addresses of the form "user@example.net" and discoverable via DNS SRV lookups on the application service name of "example.net". A certificate for this service needs to include SRV-IDs of

"_imap.example.net" and "_imaps.example.net" (see [RFC6186]) along with DNS-IDs of "example.net" and "mail.example.net". It might also include CN-IDs of "example.net" and "mail.example.net" for backward compatibility with deployed infrastructure.

Consider an SMTP Submission server at the host "submit.example.net" servicing email addresses of the form "user@example.net" and discoverable via DNS SRV lookups on the application service name of "example.net". A certificate for this service needs to include SRV-IDs of "_submission.example.net" (see [RFC6186]) along with DNS-IDs of "example.net" and "submit.example.net". It might also include CN-IDs of "example.net" and "submit.example.net" for backward compatibility with deployed infrastructure.

5. IANA Considerations

This document doesn't require any action from IANA.

6. Security Considerations

The goal of this document is to improve interoperability and thus security of email clients wishing to access email servers over TLS protected email protocols, by specifying a consistent set of rules that email service providers, email client writers and certificate authorities can use when creating server certificates.

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008.
- [RFC4409] Gellens, R. and J. Klensin, "Message Submission for Mail", RFC 4409, April 2006.
- [RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, February 2002.
- [RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.
- [RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.

- [RFC5804] Melnikov, A. and T. Martin, "A Protocol for Remotely Managing Sieve Scripts", RFC 5804, July 2010.
- [RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, March 2011.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.
- [RFC4985] Santesson, S., "Internet X.509 Public Key Infrastructure Subject Alternative Name for Expression of Service Name", RFC 4985, August 2007.

7.2. Informative References

- [RFC2595] Newman, C., "Using TLS with IMAP, POP3 and ACAP", RFC 2595, June 1999.
- [RFC6186] Daboo, C., "Use of SRV Records for Locating Email Submission/Access Services", RFC 6186, March 2011.

Appendix A. Acknowledgements

Thank you to Chris Newman for comments on this document.

The editor of this document copied lots of text from RFC 2595 and RFC 6125, so the hard work of editors of these document is appreciated.

Author's Address

Alexey Melnikov
Isode Ltd
5 Castle Business Village
36 Station Road
Hampton, Middlesex TW12 2BX
UK

EMail: Alexey.Melnikov@isode.com