

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 6, 2014

P. Hoffman
VPN Consortium
February 2, 2014

Opportunistic Encryption Using TLS
draft-hoffman-uta-opportunistic-tls-00

Abstract

This document defines the term "opportunistic encryption using TLS" as it applies to application protocols that use TLS.

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 August 6, 2014.

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.

1. Introduction

The term "opportunistic encryption" has many informal definitions, and this panoply of definitions has made discussion of using opportunistic encryption in particular protocols more difficult. The term has acquired many different meanings in different contexts, so having a single definition that can be used by protocol specifications and application developers will benefit the Internet community.

Opportunistic encryption using TLS is considered a good way to prevent passive monitoring of communications that would otherwise be sent unencrypted. It is clear that such monitoring is fairly pervasive in many Internet environments, and it is also clear that many people would like prevent their communications from being watched by governments, companies, groups, and individuals whom they do not know. Opportunistic encryption using TLS causes the start of application communication to happen later than it normally would have due to the round trips and mathematical computations required to establish a TLS session. The creators of an application program must weigh these and other factors when deciding whether or not to use opportunistic encryption in their program. Similarly, protocol designers need to take these and other factors into account when deciding whether or not to require, suggest, or even allow opportunistic encryption using TLS in their protocol specifications.

The definition of opportunistic encryption using TLS in this document explicitly sets user interface requirements for applications. Although this is rarely done in other IETF standards, doing so is required here for security reasons.

Note that "opportunistic encryption using TLS" is different than "unauthenticated TLS". The latter describes a similar but distinct concept, and it applies to different scenarios. There is a wide industry agreement that unauthenticated TLS is almost always a bad practice. The two terms are often confused, and thus "unauthenticated TLS" is described only in an appendix of this document.

This document applies to all versions of TLS, including TLS 1.2 [RFC5246], TLS 1.1 [RFC4346], and TLS 1.0 [RFC2246]. It may or may not apply to future versions of TLS. The definition of "opportunistic encryption using TLS" in this document applies to any protocol that can be protected with TLS; this means that it mostly applies to layer 7 protocols, also known as "application layer protocols". This document only defines opportunistic encryption using TLS; it does not describe opportunistic encryption with other encrypting protocols such as IPsec.

1.1. 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, BCP 14 [RFC2119].

2. Definition of 'Opportunistic Encryption Using TLS'

An application supports opportunistic encryption using TLS if the application attempts to perform TLS negotiation without the user who is running the application knowing whether or not TLS is in use. The application MUST NOT have any user-visible configuration that enables opportunistic encryption using TLS. Stated another way, it is impossible for a program to have a configuration option for opportunistic encryption: having such an option inherently is not for opportunistic encryption.

When an application that supports opportunistic encryption negotiates TLS, that application might or might not authenticate the TLS server. It is expected that the common case is that applications that supports opportunistic encryption will not authenticate the TLS servers they connect to. However, it is acceptable for an application that supports opportunistic encryption to only complete the TLS negotiation if the TLS server can be validated.

When an application that is doing opportunistic encryption successfully creates a TLS session, that application MUST NOT show the user any indication that TLS is in use.

An application that does opportunistic encryption using TLS finds the appropriate TLS server using one or more of many mechanisms, none of which are described here in detail. Some of those mechanisms include in-protocol upgrade to TLS, in-protocol pointers to TLS servers, DNS queries whose responses indicate the presence of appropriate TLS servers, and simply trying a TCP port on which TLS is expected.

3. IANA Considerations

None

4. Security Considerations

Opportunistic encryption using TLS prevents observation by passive attackers on the network. However, it doesn't completely prevent the attacker from knowing anything about the contents of the encrypted information. For example, the attacker can know what protocol is being encrypted, the approximate size of the encrypted messages, and

so on. The attacker can also learn about the cryptographic capabilities of the client and server by observing the TLS handshake.

The purpose for the requirement that the application not have any user-visible configuration that enables opportunistic encryption is that having user-visible configuration is likely to cause lower security for the Internet. A widely-used setting that says "use TLS even when it is not called for" would cause server operators to become more lax with their TLS deployments, such as not bothering to renew (or even get) widely-accepted certificates for their sites because they know that most applications could reach them with TLS anyway.

The purpose for the requirement that the application not show that TLS is in use if the TLS was established with opportunistic encryption is that such an indication is likely to cause lower security for the Internet, particularly in web browsers.

5. References

5.1. Normative References

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

5.2. Informative References

[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.

Appendix A. Unauthenticated TLS

The term "unauthenticated encryption", when used in the context of TLS, is fairly straight-forward. However, in discussions on many security and protocol mailing lists, it is often confused with "opportunistic encryption using TLS".

Unauthenticated encryption for TLS is the act of setting up a TLS session at the request of a user where the TLS client does not authenticate the TLS server.

When the TLS session is being set up at the request of the user, such as when the user enters a URL that should only be resolved with TLS, using unauthenticated TLS is rarely the expected or desired result. In such a situation, the application might allow unauthenticated TLS after giving the user some warning, or the application might even have a configuration setting that tells the application to allow unauthenticated TLS even when trying to set up an explicit TLS session.

Many security-conscious protocol developers are severely critical of applications that allow unauthenticated encryption with TLS, even if the application gives the user warnings when authentication failed. Similarly, many security-conscious protocol developers are severely critical of applications that allow unauthenticated encryption to be configured at all.

Note that "opportunistic encryption using TLS" may allow the TLS session to be set up without the client authenticating the server. This is a completely different scenario than "unauthenticated encryption" using TLS. The definition of opportunistic encryption with TLS precludes the TLS session being set up at the request of the user; the definition of unauthenticated encryption with TLS requires that the TLS session is being set up at the request of the user.

Author's Address

Paul Hoffman
VPN Consortium

Email: paul.hoffman@vpnc.org

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

Network Working Group
Internet-Draft
Updates: 1939, 3464, 3501, 5068,
6186 (if approved)
Intended status: Standards Track
Expires: February 17, 2015

K. Moore
Network Heretics
C. Newman
Oracle
August 16, 2014

Deployable Enhanced Email Privacy (DEEP)
draft-newman-email-deep-02.txt

Abstract

This specification defines a set of requirements and facilities designed to improve email privacy. This provides mechanisms intended to increase use of already deployed Transport Layer Security (TLS) technology, provide a model for mail user agents privacy assurance, and enable mail service providers to advertise improved TLS privacy facilities.

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 February 17, 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	4
2. Conventions and Terminology Used in This Document	4
3. Mail Account Privacy Assurance Level	5
3.1. High Privacy Assurance	5
3.2. Certificate Pinning	6
3.3. Low Privacy Assurance	6
3.4. Other Privacy Assurance Levels	7
4. Implicit TLS	7
4.1. Implicit TLS for POP	7
4.2. Implicit TLS for IMAP	8
4.3. Implicit TLS for SMTP Submission	8
4.4. Implicit TLS Connection Closure for POP, IMAP and SMTP	8
5. Email Security Upgrading Using Security Latches	9
5.1. Email Security Tags	9
5.2. Initial Set of Email Security Tags	10
5.3. Server DEEP Status	10
5.4. Email Security Tag Latch Failures	11
6. Recording TLS Cipher Suite in Received Header	11
7. Extensions for DEEP Status and Reporting	12
7.1. IMAP DEEP Extension	12
7.2. POP DEEP Extension	14
7.3. SMTP DEEP Extension	15
7.4. SMTP Error Extension	16
8. Use of SRV records in Establishing Configuration	16
9. Implementation Requirements	17
9.1. All Implementations (Client and Server)	17
9.1.1. Client Certificate Authentication	18
9.2. Mail Server Implementation Requirements	18
9.3. Mail User Agent Implementation Requirements	19
9.4. Non-configurable MUAs and nonstandard access protocols	20
9.5. DEEP Compliance for Anti-Virus/Anti-Spam Software and Services	20
10. Mail Service Provider Requirements	20
10.1. Server Requirements	20
10.2. MSPs MUST provide Submission Servers	20
10.3. TLS Server Certificate Requirements	21
10.4. Recommended DNS records for mail protocol servers	21
10.4.1. MX records	21
10.4.2. SRV records	21
10.4.3. TLSA records	22
10.4.4. DNSSEC	22

10.5.	MSP Server Monitoring	22
10.6.	Advertisement of DEEP status	22
10.7.	Require TLS	22
11.	IANA Considerations	22
11.1.	Security Tag Registry	22
11.2.	Initial Set of Security Tags	23
11.3.	POP3S Port Registration Update	25
11.4.	IMAPS Port Registration Update	25
11.5.	Submissions Port Registration	26
11.6.	DEEP IMAP Capability	27
11.7.	DEEP POP3 Capability	27
11.8.	DEEP SMTP EHLO Keyword	27
11.9.	SMTP Enhanced Status Code	27
11.10.	MAIL Parameters Additional-registered-clauses Sub-Registry	28
12.	Security Considerations	28
13.	References	29
13.1.	Normative References	29
13.2.	Informative References	30
Appendix A.	Design Considerations	31
Appendix B.	Open Issues	32
Appendix C.	Change Log	34
Appendix D.	Acknowledgements	35
Authors'	Addresses	35

1. Introduction

Software that provides email service via Internet Message Access Protocol (IMAP) [RFC3501], Post Office Protocol (POP) [RFC1939] and/or Simple Mail Transfer Protocol (SMTP) [RFC5321] usually has Transport Layer Security (TLS) [RFC5246] support but often does not use it in a way that maximizes end-user privacy. This specification proposes changes to email software and deployments intended to increase the use of TLS and record when that use occurs.

In brief, this memo now recommends that:

- o MUAs associate a privacy assurance level with each mail account, and the default privacy level requires use of TLS with certificate validation for all TCP connections;
- o TLS on a well-known port ("Implicit TLS") be supported for IMAP, POP, and SMTP Submission [RFC6409] for all electronic mail user agents (MUAs), servers, and service providers;
- o MUAs and mail protocol servers cooperate (via mechanisms defined in this specification) to upgrade security/privacy feature use and record/indicate that usage appropriately.

Improved use of TLS with SMTP for message relaying is described in a separate document [I-D.ietf-dane-smtp-with-dane].

The recommendations in this memo do not replace the functionality of, and are not intended as a substitute for, end-to-end encryption of electronic mail.

This draft is subject to change. Implementation of this proposal is not recommended at this time. Please discuss this proposal on the ietf-uta mailing list.

2. Conventions and Terminology 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].

This specification expresses syntax using the Augmented Backus-Naur Form (ABNF) as described in [RFC5234], including the core rules in Appendix B and rules from [RFC5322].

In examples, "C:" and "S:" indicate lines sent by the client and server respectively. If a single "C:" or "S:" label applies to

multiple lines, then the line breaks between those lines are for editorial clarity only and are not part of the actual protocol exchange.

3. Mail Account Privacy Assurance Level

The configuration necessary for a mail account includes an email address, connection information and authentication credentials for at least one mail access server (IMAP or POP) and at least one SMTP submission server. A mail user agent (MUA) typically supports one or more mail account configurations. MUAs compliant with this specification MUST associate a privacy assurance level with each mail account. MUAs MUST implement a high privacy level as described in the next section.

MUAs SHOULD continuously indicate to the user the privacy for an account's connections (e.g., via a lock icon, background colors and indications similar to those commonly used in web browsers for this purpose). Note that this could be higher than the level set at account configuration but never lower. If multiple active connections are associated with an account or view, the indication should match the privacy level provided by the least private connection.

Account configuration occurs when an MUA is first used to access a particular service, when a user wishes to access or submit mail through servers in addition to those specified or found during first use, or when a user explicitly requests to change account configuration parameters such as server names, user names, passwords, client certificates, etc. Account configuration can be entirely manual (entering server names explicitly) or partially automated via a mechanism such as DNS SRV records [RFC6186]. MUAs SHOULD use the high privacy assurance level as the default for newly configured accounts.

3.1. High Privacy Assurance

A mail account has a high privacy assurance when the following conditions are met on all TCP server connections associated with an account. This includes connections to POP, IMAP and SMTP submission servers as well as any other associated protocols defined now or in the future. Examples of protocols associated with a mail account include managesieve [RFC5804] and MTQP [RFC3887].

- o TCP connections MUST attempt to negotiate TLS via either Implicit TLS Section 4 or STARTTLS.

- o MUAs MUST implement [I-D.melnikov-email-tls-certs] and PKIX [RFC5280].
- o MUAs MAY implement DANE [RFC6698].
- o User agents MUST abort a TLS session if the TLS negotiation fails or the server's certificate or identity fails to verify. A user may reconfigure the account to lower the expected level of privacy if he/she chooses. Reduction of expected account privacy MUST NOT be done on a click-through basis.

The end user is part of the system that protects the user's privacy and security. As a result, it's critical not to present the end user with a simple action that reduces their privacy in response to certificate validation failure. An MUA which offers a user actions such as "connect anyway", "trust certificate for future connections" or "lower privacy assurance for this account" in response to certificate validation failure is not providing a high privacy assurance as defined in this section and thus does not comply with this document. Examples of acceptable actions to offer would be "work offline", "try again later", and "open service provider status web page".

3.2. Certificate Pinning

MUAs MAY implement certificate pinning as part of account setup, but MUST NOT offer this as an option in response to a failed certificate validation for an existing account. Certificate pinning occurs when the user agent saves a server certificate with the account settings and trusts that certificate for subsequent connections to that server. An MUA that allows certificate pinning MUST NOT allow a certificate pinned for one account to validate connections for other accounts.

A pinned certificate is subject to a man-in-the-middle attack at account setup time, and lacks a mechanism to revoke or securely refresh the certificate. Therefore use of a pinned certificate does not provide a high privacy assurance and an MUA MUST NOT indicate a high privacy level for an account or connection using a pinned certificate.

3.3. Low Privacy Assurance

MUAs MAY implement a low privacy assurance level for accounts. At this level, the MUA MUST attempt to negotiate TLS, but MAY ignore server certificate validation failures. MUAs MAY support use of connections without TLS, but if they do they SHOULD attempt TLS first if available and MUST implement code to reconnect without TLS if TLS

negotiation fails for reasons other than certificate validity.

Note that if the TLS certificate is not successfully validated as described in Section 3.1 or a version of SSL/TLS prior to TLS 1.0 is used, the client MUST NOT present a high privacy indication for the account or connection.

3.4. Other Privacy Assurance Levels

This specification is not intended to limit experimentation and innovation with respect to user privacy. As a result more privacy assurance levels are permitted. However, levels below the "low privacy assurance" described in the previous section are discouraged and implementers are cautioned that end users may be confused by too many privacy levels.

4. Implicit TLS

Previous standards for use of email protocols with TLS used the STARTTLS mechanism: [RFC2595], [RFC3207], and [RFC3501]. With STARTTLS, the client establishes a clear text application session and determines whether to issue a STARTTLS command based on server capabilities and client configuration. If the client issues a STARTTLS command, a TLS handshake follows that can upgrade the connection. While this mechanism has been deployed, an alternate mechanism where TLS is negotiated immediately at connection start on a separate port (referred to in this document as "Implicit TLS") has been deployed more successfully. To increase use of TLS, this specification recommends use of implicit TLS by new POP, IMAP and SMTP Submission software.

4.1. Implicit TLS for POP

When a TCP connection is established for the "pop3s" service (default port 995), a TLS handshake begins immediately. Clients MUST implement the certificate validation mechanism described in [I-D.melnikov-email-tls-certs]. Once the TLS session is established, POP3 [RFC1939] protocol messages are exchanged as TLS application data for the remainder of the TCP connection. After the server sends a +OK greeting, the server and client MUST enter AUTHORIZATION state, even if client credentials were supplied during the TLS handshake.

See Section 9.1.1 for additional information on client certificate authentication. See Section 11.3 for port registration information.

4.2. Implicit TLS for IMAP

When a TCP connection is established for the "imaps" service (default port 993), a TLS handshake begins immediately. Clients **MUST** implement the certificate validation mechanism described in [RFC3501] and **SHOULD** implement the certificate validation mechanism described in [I-D.melnikov-email-tls-certs]. Once the TLS session is established, IMAP [RFC3501] protocol messages are exchanged as TLS application data for the remainder of the TCP connection. If client credentials were provided during the TLS handshake that the server finds acceptable, the server **MAY** issue a PREAUTH greeting in which case both the server and client enter AUTHENTICATED state. If the server issues an OK greeting then both server and client enter NOT AUTHENTICATED state.

See Section 9.1.1 for additional information on client certificate authentication. See Section 11.4 for port registration information.

4.3. Implicit TLS for SMTP Submission

When a TCP connection is established for the "submissions" service (default port 465), a TLS handshake begins immediately. Clients **MUST** implement the certificate validation mechanism described in [I-D.melnikov-email-tls-certs]. Once a TLS session is established, message submission protocol data [RFC6409] is exchanged as TLS application data for the remainder of the TCP connection. (Note: the "submissions" service name is defined in section 10.3 of this document, and follows the usual convention that the name of a service layered on top of Implicit TLS consists of the name of the service as used without TLS, with an "s" appended.)

Note that the submissions port provides access to a Mail Submission Agent (MSA) as defined in [RFC6409] so requirements and recommendations for MSAs in that document apply to the submissions port, including the requirement to implement SMTP AUTH [RFC4954].

See Section 9.1.1 for additional information on client certificate authentication. See Section 11.5 for port registration information.

4.4. Implicit TLS Connection Closure for POP, IMAP and SMTP

When a client or server wishes to close the connection, it **SHOULD** initiate the exchange of TLS close alerts before TCP connection termination. The client **MAY**, after sending a TLS close alert, gracefully close the TCP connection without waiting for a TLS response from the server.

5. Email Security Upgrading Using Security Latches

Once an improved email security or privacy mechanism is deployed and ready for general use, it is desirable to continue using it for all future email service. For example, TLS is widely deployed in email software, but use of TLS is often not required. At the time this is written, deployed mail user agents (MUAs) [RFC5598] usually make a determination if TLS is available when an account is first configured and may require use of TLS with that account if and only if it was initially available. If the service provider makes TLS available after initial client configuration, many MUAs will not notice the change.

Alternatively, a security feature may be purely opportunistic and thus subject to downgrade attacks. For example, at the time this was written, most TLS stacks that support TLS 1.2 will fallback to TLS 1.0 without alerting the client of the reduced security. Thus a variety of active attacks could cause the loss of TLS 1.2 benefits. Only if client policy is upgraded to require TLS 1.2 can the client prevent all downgrade attacks. However, this sort of security policy upgrade will be ignored by most users unless it is automated.

This section describes a mechanism, called "security latches", which is designed to permit an MUA to recognize when a service provider has committed to provide certain server security features, and that it's safe for the client to change its configuration for that account to require that such features be present in future sessions with that server. When an MUA implements both privacy assurance levels and security latches, then both the end-user and the service provider independently have the ability to improve the end-user's privacy.

Note that security latches are a mechanism similar to HTTP Strict Transport Security (HSTS) [RFC6797] but are extensible.

5.1. Email Security Tags

Each security latch is given a name known as an email security tag. An email security tag is a short alphanumeric token that represents a security facility that can be used by an IMAP, POP or SMTP Submission session. When a server advertises a security tag it is making a commitment to support that security facility indefinitely and recommending that the client save that security tag with the account configuration and require that security feature for future connections to that server. When a security tag is saved by the client in this way, it is then considered latched. For the "tls10" and/or "tls12" tags, the client SHOULD refuse to connect to the server unless the appropriate level of TLS is successfully negotiated. If these tags are still advertised by the server after

negotiation, the client SHOULD latch these tags. Other security tags are latched if they are advertised by the server, TLS is active and the client successfully authenticates the server with the TLS session. Once a security tag is latched, all subsequent connections to that host require that security feature. For this privacy protection to work as desired clients MUST NOT offer a click-through-to-connect action when unable to achieve connection security matching the latched security tags.

An identifier for a security tag has the following formal syntax:

```
security-tag = ALPHA *63(ALPHA / DIGIT / "-" / "_")
```

5.2. Initial Set of Email Security Tags

This section describes an initial set of email security tags. The IANA Considerations Section 11 defines a registry so that more tags can be defined in the future. The initial set of tags are defined in Section 11.2 and include tls10, tls12, tls-cert and tls-dane-tlsa.

5.3. Server DEEP Status

Servers supporting this extension MUST advertise a DEEP status. This status includes a list of security-tags the server administrator has explicitly configured as recommended for use by end-users (the list MAY be empty), an optional https Uniform Resource Locator (URL) [RFC2818] that the client can save and subsequently resolve for the user in the event of a security connection problem, and the DEEP status can be extended by future updates to this specification. DEEP status has the following formal syntax:

```
EXTCHAR      = 0x20-21 / 0x23-2E / 0x30-3B / 0x3D-40
               / 0x5B-60 / 0x7B-7E
               ; printable characters excluding " \ < and ALPHA

deep-extend   = EXTCHAR *(EXTCHAR / ALPHA / "<")
               ; clients MUST ignore, for future extensibility

deep-status   = [deep-tag *(SP deep-tag)]

deep-tag      = deep-https / security-tag / deep-extend

deep-https    = "<" <URI from RFC 3986 with https scheme> ">"
```

The syntax for a Uniform Resource Identifier (URI) is defined in [RFC3986]. Protocol extensions to advertise DEEP status are defined in Section 7.

If the client successfully negotiates TLS and authenticates the server (e.g., via `tls-cert`, `tls-dane-tlsa` or `SCRAM-SHA1-PLUS` with channel bindings [RFC5802]), then the client SHOULD record the server's DEEP status information in the account configuration with the server's hostname. Otherwise, the client SHOULD ignore the server-provided DEEP status except for the `"tls10"` and `"tls12"` security tags.

5.4. Email Security Tag Latch Failures

When a security tag latch has been set for connections from a client to a server and the property identified by that tag is no longer available, this results in a connection failure. An MUA SHOULD inform the user of a potential threat to their privacy and offer to resolve a previously-recorded DEEP status https URL if one is available. An MUA might suggest deleting the account and re-creating it as a cumbersome mechanism to reset the latches. MUAs are discouraged from offering a lightweight option to reset or ignore latches as this defeats the privacy benefit they provide to end users.

6. Recording TLS Cipher Suite in Received Header

The ESMTPS transmission type [RFC3848] provides trace information that can indicate TLS was used when transferring mail. However, TLS usage by itself is not a guarantee of privacy or security. The TLS cipher suite provides additional information about the level of privacy or security made available for a connection. This defines a new SMTP `"tls"` Received header additional-registered-clause that is used to record the TLS cipher suite that was negotiated for the connection. The value included in this additional clause SHOULD be the registered cipher suite name (e.g., `TLS_DHE_RSA_WITH_AES_128_CBC_SHA`) included in the TLS cipher suite registry. In the event the implementation does not know the name of the cipher suite (a situation that should be remedied promptly), a four-digit hexadecimal cipher suite identifier MAY be used. The ABNF for the field follows:

```
tls-cipher-clause = CFWS "tls" FWS tls-cipher

tls-cipher        = tls-cipher-suite-name / tls-cipher-suite-hex

tls-cipher-name   = ALPHA *(ALPHA / DIGIT / "_")
                  ; as registered in IANA cipher suite registry

tls-cipher-hex    = "0x" 4HEXDIG
```

7. Extensions for DEEP Status and Reporting

This memo defines optional mechanisms for use by MUAs to communicate DEEP status to servers. One purpose of such mechanisms is to permit servers to determine which and how many clients have latched security facilities, and thus, to permit operators to be aware of potential impact to their users should support for such facilities be changed. For IMAP, the existing ID command is extended to provide this capability. For SMTP Submission, a new CLIENT command is defined. No similar mechanism is defined for POP in this version of the memo to keep POP simpler, but one may be added in the future if deemed necessary.

In addition, for each of IMAP, POP, and SMTP, a new DEEP capability is defined so the client can access the DEEP status.

7.1. IMAP DEEP Extension

When an IMAP server advertises the DEEP capability, that indicates the IMAP server implements IMAP4 ID [RFC2971] with additional field values defined here. This is grouped with the ID command because that is the existing IMAP mechanism for clients to report data for server logging, and provides a way for the server to report the DEEP status.

deep From server to client, the argument to this ID field is the server DEEP status. Servers **MUST** provide this information in response to an ID command.

latch From client to server, this is a space-separated list of security tags the client has latched for this server. Servers **MAY** record this information so administrators know the expected privacy level of the client and can thus act to avoid security latch failures (e.g., by renewing server certificates on time, etc).

latch-fail From client to server, a space-separated list including one or more security tag the client has latched that the client was unable to achieve. This allows clients to report errors to the server prior to terminating the connection to the server in the event an acceptable privacy level is unavailable.

security-tags From client to server, this is a space-separated list of security tags the client supports that are not latched.

tls Server-side IMAP proxies that accept TLS connections from clients and connect in-the-clear over a fully private secure network to the server SHOULD use this field to report the tls-cipher (syntax as defined in Section 6) to the server.

IMAP clients SHOULD use the IMAP ID command to report latch failures and determine the server DEEP status. Clients MAY use the ID command to report other latch or security tag information. IMAP servers MUST implement the ID command at least to report DEEP status to clients.

```
<client connected to port 993 and negotiated TLS successfully>
S: * OK [CAPABILITY IMAP4rev1 DEEP ID AUTH=PLAIN
    AUTH=SCRAM-SHA-1] hello
C: a001 ID ("name" "Demo Mail" "version" "1.5" "latch"
    "tls10 tls-cert" "security-tags" "tls12")
S: * ID ("name" "Demo Server" "version" "1.7" "deep-status"
    "<https://www.example.com/privacy-support.html>")
S: a001 OK ID completed
```

Example 1

This example shows a client that successfully negotiated TLS version 1.0 or later and verified the server's certificate as required by IMAP. The client supports TLS 1.2. However, even if the client successfully negotiated TLS 1.2, it will not latch that security tag automatically because the server did not advertise that tag. If the client successfully validated the server certificate, it will latch the provided URL.

```
<client connected to port 993 and negotiated TLS successfully>
S: * OK [CAPABILITY IMAP4rev1 DEEP ID AUTH=PLAIN
    AUTH=SCRAM-SHA-1] hello
C: a001 ID ("name" "Demo Mail" "version" "1.5" "latch-failure"
    "tls-cert")
S: * ID ("name" "Demo Server" "version" "1.7" "deep-status"
    "tls10 <https://www.example.com/privacy-support.html>")
S: a001 OK ID completed
C: a002 LOGOUT
```

Example 2

This example shows a client that negotiated TLS, but was unable to verify the server's certificate. The latch-failure informs the server of this problem, at which point the client can disconnect. If the client had previously latched a URI for privacy problems from this server, it could offer to resolve that URI. However, the deep-status in this exchange is ignored due to the latch failure.


```

<IMAP Proxy connected over private network on port 143, there is
a client connected to the proxy on port 993 that negotiated TLS>
S: * OK [CAPABILITY IMAP4rev1 DEEP ID AUTH=PLAIN
    AUTH=SCRAM-SHA-1] hello
C: a001 ID ("name" "Demo Mail" "version" "1.5" "latch"
    "tls10 tls-cert" "security-tags" "tls12"
    "tls" "TLS_RSA_WITH_AES_128_CBC_SHA")
S: * ID ("name" "Demo Server" "version" "1.7" "deep-status"
    "tls10 tls-cert <https://www.example.com/support.html>")
S: a001 OK ID completed

```

Example 3

This example shows the connection from an IMAP proxy to a back-end server. The client connected to the proxy and sent the ID command shown in example 1, and the proxy has added the "tls" item to the ID command so the back-end server can log the cipher suite that was used on the connection from the client.

7.2. POP DEEP Extension

POP servers supporting this specification MUST implement the POP3 extension mechanism [RFC2449]. POP servers MUST advertise the DEEP capability with an argument indicating the server's DEEP status.

```

<client connected to port 995 and negotiated TLS successfully>
S: +OK POP server ready
C: CAPA
S: +OK Capability list follows
S: TOP
S: SASL PLAIN SCRAM-SHA-1
S: RESP-CODES
S: PIPELINING
S: UIDL
S: DEEP tls10 tls12 <https://www.example.com/privacy-support.html>
S: .

```

Example

After verifying the TLS server certificate and issuing CAPA, the client can latch any or all of the DEEP status. If the client connects to this same server later and has a privacy failure, the client can direct the user's browser to the previously-latched URI where the service provider may provide advice to the end user.

7.3. SMTP DEEP Extension

SMTP Submission servers supporting this specification MUST implement the DEEP SMTP extension. The name of this extension is DEEP. The EHLO keyword value is DEEP and the deep-status ABNF is the syntax of the EHLO keyword parameters. This does not add parameters to the MAIL FROM or RCPT TO commands. This also adds a CLIENT command to SMTP which is used to report client information to the server. The formal syntax for the command follows:

```
deep-cmd          = "CLIENT" 1*(SP deep-parameter)

deep-parameter    = name / version / latch / latch-fail
                  / security-tags / tls / future-extension

name              = "name=" esmtp-value

version           = "version=" esmtp-value

latch             = "latch=" security-tag *("," security-tag)

latch-fail        = "latch-fail=" security-tag
                  *("," security-tag)

security-tags     = "security-tags=" security-tag
                  *("," security-tag)

tls               = "tls=" tls-cipher

future-extension  = esmtp-param

esmtp-param       = <as defined in RFC 5321>

esmtp-value       = <as defined in RFC 5321>
```

The CLIENT command parameters listed here have the same meaning as the parameters used in the IMAP DEEP extension (Section 7.1). The server responds to the CLIENT command with a "250" if the command has correct syntax and a "501" if the command has incorrect syntax.

```
<client connected to port 465 and negotiated TLS successfully>
S: 220 example.com Demo SMTP Submission Server
C: EHLO client.example.com
S: 250-example.com
S: 250-8BITMIME
S: 250-PIPELINING
S: 250-DSN
S: 250-AUTH PLAIN LOGIN
S: 250-DEEP tls10 tls-cert <https://www.example.com/status.html>
S: 250-BURL imap
S: 250 SIZE 0
C: CLIENT name=demo_submit version=1.5 latch=tls10,tls-cert
    security-tags=tls12
S: 250 OK
```

Example

7.4. SMTP Error Extension

Although this document focuses on SMTP Submission, it is possible to use security latches for SMTP transport as well. When MTA transport fails due to a security latch, the MTA MUST use the SMTP enhanced status code X.7.TBD. The SMTP notary response [RFC3464] for a security latch failure MUST include an additional "SMTP-Security-Latch" recipient-specific header field that includes a space-delimited list including one or more security latch that failed. The ABNF for this new field follows:

```
CFWS                = <defined in RFC 5322>

FWS                 = <defined in RFC 5322>

smtp-security-latch = "SMTP-Security-Latch:" CFWS
                      security-tag *(FWS security-tag)
```

8. Use of SRV records in Establishing Configuration

This section updates [RFC6186] by changing the preference rules and adding a new SRV service label `_submissions._tcp` to refer to Message Submission with implicit TLS.

User-configurable MUAs SHOULD support use of [RFC6186] for account setup. However, when using configuration information obtained by this method, MUAs SHOULD default to a high privacy assurance level, unless the user has explicitly requested reduced privacy. This will have the effect of causing the MUA to ignore advertised configurations which do not support TLS, even when those advertised

configurations have a higher priority than other advertised configurations.

When using [RFC6186] configuration information, Mail User Agents SHOULD NOT automatically establish new configurations that do not require TLS for all servers, unless there are no advertised configurations using TLS. If such a configuration is chosen, prior to attempting to authenticate to the server or use the server for message submission, the MUA SHOULD warn the user that traffic to that server will not be encrypted and that it will therefore likely be intercepted by unauthorized parties. The specific wording is to be determined by the implementation, but it should adequately capture the sense of risk given the widespread incidence of mass surveillance of email traffic.

When establishing a new configuration for connecting to an IMAP, POP, or SMTP Submission server, an MUA SHOULD NOT blindly trust SRV records unless they are signed by DNSSEC and have a valid signature. Instead, the MUA SHOULD warn the user that the DNS-advertised mechanism for connecting to the server is not authenticated, and request the user to manually verify the connection details by reference to his or her mail service provider's documentation.

Similarly, an MUA MUST NOT consult SRV records to determine which servers to use on every connection attempt, unless those SRV records are signed by DNSSEC and have a valid signature. However, an MUA MAY consult SRV records from time to time to determine if an MSP's server configuration has changed, and alert the user if it appears that this has happened. This can also serve as a means to encourage users to upgrade their configurations to require TLS if and when their MSPs support it.

9. Implementation Requirements

This section details requirements for implementations of electronic mail protocol clients and servers. A requirement for a client or server implementation to support a particular feature is not the same thing as a requirement that a client or server running a conforming implementation be configured to use that feature. Requirements for Mail Service Providers (MSPs) are distinct from requirements for protocol implementations, and are listed in a separate section.

9.1. All Implementations (Client and Server)

These requirements apply to MUAs as well as POP, IMAP and SMTP Submission servers.

- o All implementations MUST be configurable to support implicit TLS using the TLS 1.2 protocol or later [RFC5246] including support for the mandatory-to-implement TLS 1.2 cipher suite TLS_RSA_WITH_AES_128_CBC_SHA.
- o IMAP implementations MUST support the IMAP4rev1 mandatory-to-implement cipher suite TLS_RSA_WITH_RC4_128_MD5 for any connections made or received via IMAP although this MAY be disabled by default.
- o All implementations MUST be configurable to require TLS before performing any operation other than capability discovery and STARTTLS.

9.1.1. Client Certificate Authentication

MUAs and mail servers MAY implement client certificate authentication on the implicit TLS port. Servers MUST NOT request a client certificate during the TLS handshake unless the server is configured to accept some client certificates as sufficient for authentication and the server has the ability to determine a mail server authorization identity matching such certificates. How to make this determination is presently implementation specific. Clients MUST NOT provide a client certificate during the TLS handshake unless the server requests one and the client has determined the certificate can be safely used with that specific server, OR the client has been explicitly configured by the user to use that particular certificate with that server. How to make this determination is presently implementation specific. If the server accepts the client's certificate as sufficient for authorization, it MUST enable the SASL EXTERNAL [RFC4422] mechanism. An IMAPS server MAY issue a PREAUTH greeting instead of enabling SASL EXTERNAL. A client supporting client certificate authentication with implicit TLS MUST implement the SASL EXTERNAL [RFC4422] mechanism using the appropriate authentication command (AUTH for POP3 [RFC5034], AUTH for SMTP Submission [RFC4954], AUTHENTICATE for IMAP [RFC3501]).

9.2. Mail Server Implementation Requirements

These requirements apply to servers that implement POP, IMAP or SMTP Submission.

- o Servers MUST implement the DEEP extension described in Section 7
- o IMAP and SMTP submission servers SHOULD implement and be configurable to support STARTTLS. This enables discovery of new TLS availability, and can increase usage of TLS by legacy clients.

- o Servers MUST NOT advertise STARTTLS if it is unlikely to succeed based on server configuration (e.g., there is no server certificate installed).
- o SMTP message submission servers that have negotiated TLS SHOULD add a Received header field to the message including the tls clause described in Section 6.
- o Servers MUST be configurable to include the TLS cipher information in any connection or user logging or auditing facility they provide.

9.3. Mail User Agent Implementation Requirements

This section describes requirements on Mail User Agents (MUAs) using IMAP, POP, and/or Submission protocols. Note: Requirements pertaining to use of Submission servers are also applicable to use of SMTP servers (e.g., port 25) for mail submission.

- o User agents SHOULD indicate at configuration time, the expected level of privacy based on appropriate security inputs such as which security latches are pre-set, the number of trust anchors, certificate validity, use of an extended validation certificate, TLS version supported, and TLS cipher suites supported by both server and client.
- o MUAs SHOULD detect when STARTTLS and/or implicit TLS becomes available for a protocol and set the tls10 latch if the server advertises that latch.
- o Whenever requested to establish any configuration that does not require both TLS and server certificate verification to talk to a server or account, an MUA SHOULD warn its user that his or her mail traffic (including password, if applicable) will be exposed to attackers, and give the user an opportunity to abort the connection prior to transmission of any such password or traffic.
- o MUAs SHOULD implement the "tls12" security latch (the TLS library has to provide an API that controls permissible TLS versions and communicates the negotiated TLS protocol version to the application for this to be possible).
- o See Section 3 for additional requirements.

9.4. Non-configurable MUAs and nonstandard access protocols

MUAs which are not configurable to use user-specified servers MUST implement TLS or similarly other strong encryption mechanism when communicating with their mail servers. This generally applies to MUAs that are pre-configured to operate with one or more specific services, whether or not supplied by the vendor of those services.

MUAs using protocols other than IMAP, POP, and Submission to communicate with mail servers, MUST implement TLS or other similarly robust encryption mechanism in conjunction with those protocols.

9.5. DEEP Compliance for Anti-Virus/Anti-Spam Software and Services

There are multiple ways to connect an Anti-Virus and/or Anti-Spam (AVAS) service to a mail server. Some mechanisms, such as the de-facto milter protocol do not impact DEEP. However, some services use an SMTP relay proxy that intercepts mail at the application layer to perform a scan and proxy to the real MTA. Deploying AVAS services in this way can cause many problems [RFC2979] including direct interference with DEEP and privacy reduction. An AVAS product or service is considered DEEP compliant if all IMAP, POP and SMTP-related software it includes is DEEP compliant and it advertises all security latches that the actual MTA advertises.

10. Mail Service Provider Requirements

This section details requirements for providers of IMAP, POP, and/or SMTP submission services, for providers who claim to conform to this specification.

10.1. Server Requirements

Mail Service Providers MUST use server implementations that conform to this specification.

10.2. MSPs MUST provide Submission Servers

This document updates the advice in [RFC5068] by making Implicit TLS on port 465 the preferred submission port.

Mail Service Providers that accept mail submissions from end-users using the Internet Protocol MUST provide one or more SMTP Submission servers for this purpose, separate from the SMTP servers used to process incoming mail. Those submission servers MUST be configured to support Implicit TLS on port 465 and SHOULD support STARTTLS if port 587 is used.

MSPs MAY also support submission of messages via one or more designated SMTP servers to facilitate compatibility with legacy MUAs.

Discussion: SMTP servers used to accept incoming mail or to relay mail are expected to accept mail in cleartext. This is incompatible with the purpose of this memo which is to encourage encryption of traffic between mail servers. There is no such requirement for mail submission servers to accept mail in cleartext or without authentication. For other reasons, use of separate SMTP submission servers has been best practice for many years.

10.3. TLS Server Certificate Requirements

MSPs MUST maintain valid server certificates for all servers. Those server certificates SHOULD present DNS-IDs and SRV-IDs conforming to [RFC6125] and which will be recognized by MUAs meeting the requirements of that specification. In addition, those server certificates MAY provide other DNS-IDs, SRV-IDs, or CN-IDs needed for compatibility with existing MUAs.

If a protocol server provides service for more than one mail domain, it MAY use a separate IP address for each domain and/or a server certificates that advertises multiple domains. This will generally be necessary unless and until it is acceptable to impose the constraint that the server and all clients support the Server Name Indication extension to TLS [RFC6066].

10.4. Recommended DNS records for mail protocol servers

This section discusses not only the DNS records that are recommended, but also implications of DNS records for server configuration and TLS server certificates.

10.4.1. MX records

It is recommended that MSPs advertise MX records for handling of inbound mail (instead of relying entirely on A or AAAA records), and that those MX records be signed using DNSSEC. This is mentioned here only for completeness, as handling of inbound mail is out of scope for this document.

10.4.2. SRV records

MSPs SHOULD advertise SRV records to aid MUAs in determination of proper configuration of servers, per the instructions in [RFC6186].

MSPs SHOULD advertise servers that support Implicit TLS in preference to those which support cleartext and/or STARTTLS operation.

10.4.3. TLSA records

MSPs SHOULD advertise TLSA records to provide an additional trust anchor for public keys used in TLS server certificates. However, TLSA records MUST NOT be advertised unless they are signed using DNSSEC.

10.4.4. DNSSEC

All DNS records advertised by an MSP as a means of aiding clients in communicating with the MSP's servers, SHOULD be signed using DNSSEC.

10.5. MSP Server Monitoring

MSPs SHOULD regularly and frequently monitor their various servers to make sure that: TLS server certificates remain valid and are not about to expire, TLSA records match the public keys advertised in server certificates, are signed using DNSSEC, server configurations are consistent with SRV advertisements, and DNSSEC signatures are valid and verifiable. Failure to detect expired certificates and DNS configuration errors in a timely fashion can result in significant loss of service for an MSP's users and a significant support burden for the MSP.

10.6. Advertisement of DEEP status

MSPs SHOULD advertise a DEEP status that includes `tls10`, `tls-cert` and an HTTPS URL that can be used to inform clients of service outages or problems impacting client privacy. Note that advertising `tls-cert` is a commitment to maintain and renew server certificates.

10.7. Require TLS

New servers and services SHOULD be configured to require TLS unless it's necessary to support legacy clients or existing client configurations.

11. IANA Considerations

11.1. Security Tag Registry

IANA shall create (has created) the registry "Email Security Tags". This registry is a single table and will use an expert review process [RFC5226]. Each registration will contain the following fields:

Name: The name of the security tag. This follows the security-tag ABNF.

Description: This describes the meaning of the security tag and the conditions under which the tag is latched.

Intended Usage: One of COMMON, LIMITED USE or OBSOLETE.

Reference: Optional reference to specification.

Submitter: The identify of the submitter or submitters.

Change Controller: The identity of the change controller for the registration. This will be "IESG" in case of registrations in IETF-produced documents.

The expert reviewer will verify the tag name follows the ABNF, and that the description field is clear, unambiguous, does not overlap existing deployed technology, does not create security or privacy problems and appropriately considers interoperability issues. Email security tags intended for LIMITED USE have a lower review bar (interoperability and overlap issues are less of a concern). The reviewer may approve a registration, reject for a stated reason or recommend the proposal have standards track review due to importance or difficult subtleties.

Standards-track registrations may be updated if the relevant standards are updated as a consequence of that action. Non-standards-track entries may be updated by the listed change controller. The entry's name and submitter may not be changed. In exceptional cases, any aspect of any registered entity may be updated at the direction of the IESG (for example, to correct a conflict).

11.2. Initial Set of Security Tags

This document defines four initial security tags for the security tag registry as follows:

Name: tls10

Description: This indicates TLS version 1.0 [RFC2246] or later was negotiated successfully including negotiation of a strong encryption layer with a symmetric key of at least 128 bits. This tag does not indicate the server certificate was valid. This tag is latched if the client sees this tag in the advertised server DEEP status provided after successfully negotiating TLS version 1.0 or later.

Intended Usage: COMMON

Reference: RFC XXXX (this document once published)

Submitter: Authors of this document

Change Controller: IESG

Name: tls12

Description: This indicates TLS version 1.2 [RFC5246] or later was negotiated successfully including negotiation of a strong encryption layer with a symmetric key of at least 128 bits. This tag does not indicate the server certificate was valid. This tag is latched if the client sees this tag in the advertised server DEEP status provided after successfully negotiating TLS version 1.2 or later.

Intended Usage: COMMON

Reference: RFC XXXX (this document once published)

Submitter: Authors of this document

Change Controller: IESG

Name: tls-cert

Description: This tag indicates that TLS was successfully negotiated and the server certificate was successfully verified by the client using PKIX [RFC5280] and the server certificate identity was verified using the algorithm appropriate for the protocol (see Section 4). This tag is latched if the client sees this tag in the advertised server DEEP status after successfully negotiating TLS and verifying the certificate and server identity.

Intended Usage: COMMON

Reference: RFC XXXX (this document once published)

Submitter: Authors of this document

Change Controller: IESG

Name: tls-dane-tlsa

Description: This tag indicates that TLS was successfully negotiated and the server certificate was successfully verified by the client using the procedures described in [RFC6698] and the server certificate identity was verified using the algorithm appropriate for the protocol (see Section 4). This tag is latched if the client sees this tag in the advertised server DEEP status after successfully negotiating TLS and verifying the certificate and server identity.

Intended Usage: COMMON

Reference: RFC XXXX (this document once published)

Submitter: Authors of this document

Change Controller: IESG

11.3. POP3S Port Registration Update

IANA is asked to update the registration of the TCP well-known port 995 using the following template ([RFC6335]):

Service Name: pop3s
Transport Protocol: TCP
Assignee: IETF <iesg@ietf.org>
Contact: IESG <iesg@ietf.org>
Description: POP3 over TLS protocol
Reference: RFC XXXX (this document once published)

Service Name: pop3s
Transport Protocol: TCP
Assignee: IETF <iesg@ietf.org>
Contact: IESG <iesg@ietf.org>
Description: POP3 over TLS protocol
Reference: RFC XXXX (this document once published)

11.4. IMAPS Port Registration Update

IANA is asked to update the registration of the TCP well-known port 993 using the following template ([RFC6335]):

Service Name: imaps
Transport Protocol: TCP
Assignee: IETF <iesg@ietf.org>
Contact: IESG <iesg@ietf.org>
Description: IMAP over TLS protocol

Reference: RFC XXXX (this document once published)

Service Name: imaps
Transport Protocol: TCP
Assignee: IETF <iesg@ietf.org>
Contact: IESG <iesg@ietf.org>
Description: IMAP over TLS protocol
Reference: RFC XXXX (this document once published)

11.5. Submissions Port Registration

IANA is asked to assign an alternate usage of port 465 in addition to the current assignment using the following template ([RFC6335]):

Service Name: submissions
Transport Protocol: TCP
Assignee: IETF <iesg@ietf.org>
Contact: IESG <iesg@ietf.org>
Description: Message Submission over TLS protocol
Reference: RFC XXXX (this document once published)

Service Name: submissions
Transport Protocol: TCP
Assignee: IETF <iesg@ietf.org>
Contact: IESG <iesg@ietf.org>
Description: Message Submission over TLS protocol
Reference: RFC XXXX (this document once published)

This is a one time procedural exception to the rules in RFC 6335. This requires explicit IESG approval and does not set a precedent. Historically, port 465 was briefly registered as the "smtps" port. This registration made no sense as the SMTP transport MX infrastructure has no way to specify a port so port 25 is always used. As a result, the registration was revoked and was subsequently reassigned to a different service. In hindsight, the "smtps" registration should have been renamed or reserved rather than revoked. Unfortunately, some widely deployed mail software interpreted "smtps" as "submissions" [RFC6409] and used that port for email submission by default when an end-user requests security during account setup. If a new port is assigned for the submissions service, email software will either continue with unregistered use of port 465 (leaving the port registry inaccurate relative to de-facto practice and wasting a well-known port), or confusion between the de-facto and registered ports will cause harmful interoperability problems that will deter use of TLS for message submission. The authors believe both of these outcomes are less desirable than a wart in the registry documenting real-world usage of a port for two purposes. Although STARTTLS-on-port-587 has deployed, it has not

replaced deployed use of implicit TLS submission on port 465.

11.6. DEEP IMAP Capability

This document adds the DEEP capability to the IMAP capabilities registry. This is described in Section 7.1.

11.7. DEEP POP3 Capability

This document adds the DEEP capability to the POP3 capabilities registry.

CAPA Tag: DEEP

Arguments: deep-status

Added Commands: none

Standard Commands affected: none

Announced status / possible differences: both / may change after STLS

Commands Valid in States: N/A

Specification Reference: This document

Discussion: See Section 7.2.

11.8. DEEP SMTP EHLO Keyword

This document adds the DEEP EHLO Keyword to the SMTP Service Extension registry. This is described in Section 7.3.

11.9. SMTP Enhanced Status Code

This document adds the following entry to the "SMTP Enhanced Status Codes" registry created by [RFC5248].

Code: X.7.TBD (IANA, please assign the next available number)

Sample Text: Message Transport Failed due to missing required security.

Associated Basic Status Code: 450, 454, 550, 554

Description This code indicates an SMTP server was unable to forward a message to the next host necessary for delivery because it required a higher level of transport security or privacy than was available. The temporary form of this error is preferred in case the problem is caused by a temporary administrative error such as an expired server certificate.

Reference This document

Submitter C. Newman

Change Controller IESG

11.10. MAIL Parameters Additional-registered-clauses Sub-Registry

This document adds the following entry to the "Additional-registered-clauses" sub-registry of the "MAIL Parameters" registry, created by [RFC5321]:

Clause Name: tls

Description: Indicates the TLS cipher suite used for a transport connection.

Syntax Summary: See tls-cipher ABNF Section 6

Reference: This document.

12. Security Considerations

This entire document is about security considerations. In general, this is targeted to improve mail privacy and to mitigate threats external to the email system such as network-level snooping or interception; this is not intended to mitigate active attackers who have compromised service provider systems.

It could be argued that sharing the name and version of the client software with the server has privacy implications. Although providing this information is not required, it is encouraged so that mail service providers can more effectively inform end-users running old clients that they need to upgrade to protect their privacy, or know which clients to use in a test deployment prior to upgrading a server to have higher security requirements.

13. References

13.1. Normative References

- [RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2449] Gellens, R., Newman, C., and L. Lundblade, "POP3 Extension Mechanism", RFC 2449, November 1998.
- [RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.
- [RFC2971] Showalter, T., "IMAP4 ID extension", RFC 2971, October 2000.
- [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.
- [RFC3464] Moore, K. and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", RFC 3464, January 2003.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
- [RFC5034] Siemborski, R. and A. Menon-Sen, "The Post Office Protocol (POP3) Simple Authentication and Security Layer (SASL) Authentication Mechanism", RFC 5034, July 2007.
- [RFC5068] Hutzler, C., Crocker, D., Resnick, P., Allman, E., and T. Finch, "Email Submission Operations: Access and Accountability Requirements", BCP 134, RFC 5068, November 2007.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.
- [RFC5248] Hansen, T. and J. Klensin, "A Registry for SMTP Enhanced Mail System Status Codes", BCP 138, RFC 5248, June 2008.

- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [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.
- [RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.
- [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.
- [RFC6186] Daboo, C., "Use of SRV Records for Locating Email Submission/Access Services", RFC 6186, March 2011.
- [RFC6409] Gellens, R. and J. Klensin, "Message Submission for Mail", STD 72, RFC 6409, November 2011.
- [I-D.melnikov-email-tls-certs] Melnikov, A., "Updated TLS Server Identity Check Procedure for Email Related Protocols", draft-melnikov-email-tls-certs-01 (work in progress), October 2013.
- [I-D.ietf-dane-smtp-with-dane] Dukhovni, V. and W. Hardaker, "SMTP security via opportunistic DANE TLS", draft-ietf-dane-smtp-with-dane-02 (work in progress), October 2013.

13.2. Informative References

- [RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.
- [RFC2595] Newman, C., "Using TLS with IMAP, POP3 and ACAP", RFC 2595, June 1999.
- [RFC2979] Freed, N., "Behavior of and Requirements for Internet Firewalls", RFC 2979, October 2000.

- [RFC3848] Newman, C., "ESMTP and LMTP Transmission Types Registration", RFC 3848, July 2004.
- [RFC3887] Hansen, T., "Message Tracking Query Protocol", RFC 3887, September 2004.
- [RFC4422] Melnikov, A. and K. Zeilenga, "Simple Authentication and Security Layer (SASL)", RFC 4422, June 2006.
- [RFC4954] Siemborski, R. and A. Melnikov, "SMTP Service Extension for Authentication", RFC 4954, July 2007.
- [RFC5598] Crocker, D., "Internet Mail Architecture", RFC 5598, July 2009.
- [RFC5802] Newman, C., Menon-Sen, A., Melnikov, A., and N. Williams, "Salted Challenge Response Authentication Mechanism (SCRAM) SASL and GSS-API Mechanisms", RFC 5802, July 2010.
- [RFC5804] Melnikov, A. and T. Martin, "A Protocol for Remotely Managing Sieve Scripts", RFC 5804, July 2010.
- [RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.
- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, August 2011.
- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.
- [RFC6797] Hodges, J., Jackson, C., and A. Barth, "HTTP Strict Transport Security (HSTS)", RFC 6797, November 2012.

Appendix A. Design Considerations

This section is not normative.

The first version of this was written independently from draft-moore-email-tls-00.txt; subsequent versions merge ideas from both drafts.

One author of this document was also the author of RFC 2595 that

became the standard for TLS usage with POP and IMAP, and the other author was perhaps the first to propose that idea. In hindsight both authors now believe that that approach was a mistake. At this point the authors believe that while anything that makes it easier to deploy TLS is good, the desirable end state is that these protocols always use TLS, leaving no need for a separate port for cleartext operation except to support legacy clients while they continue to be used. The separate port model for TLS is inherently simpler to implement, debug and deploy. It also enables a "generic TLS load-balancer" that accepts secure client connections for arbitrary foo-over-TLS protocols and forwards them to a server that may or may not support TLS. Such load-balancers cause many problems because they violate the end-to-end principle and the server loses the ability to log security-relevant information about the client unless the protocol is designed to forward that information (as this specification does for the cipher suite). However, they can result in TLS deployment where it would not otherwise happen which is a sufficiently important goal that it overrides the problems.

Although STARTTLS appears only slightly more complex than separate-port TLS, we again learned the lesson that complexity is the enemy of security in the form of the STARTTLS command injection vulnerability (CERT vulnerability ID #555316). Although there's nothing inherently wrong with STARTTLS, the fact it resulted in a common implementation error (made independently by multiple implementers) suggests it is a less secure architecture than Implicit TLS.

Section 7 of RFC 2595 critiques the separate-port approach to TLS. The first bullet was a correct critique. There are proposals in the http community to address that, and use of SRV records as described in RFC 6186 resolves that critique for email. The second bullet is correct as well, but not very important because useful deployment of security layers other than TLS in email is small enough to be effectively irrelevant. The third bullet is incorrect because it misses the desirable option of "use and latch-on TLS if available". The fourth bullet may be correct, but is not a problem yet with current port consumption rates. The fundamental error was prioritizing a perceived better design based on a mostly valid critique over real-world deployability. But getting security and privacy facilities actually deployed is so important it should trump design purity considerations.

Appendix B. Open Issues

There are many open issues with this document. Here is an attempt to enumerate some of them:

- o Port 465 is presently used for two purposes: for submissions by a large number of clients and service providers and for the "urd" protocol by one vendor. Actually documenting this current state is controversial as discussed in the IANA considerations section. However, there is no good alternative. Registering a new port for submissions when port 465 is widely used for that purpose already will just create interoperability problems. Registering a port that's only used if advertised by an SRV record (RFC 6186) would not create interoperability problems but would require all client and server deployments and software to change significantly which is contrary to the goal of promoting more TLS use. Encouraging use of STARTTLS on port 587 would not create interoperability problems, but is unlikely to have impact on current undocumented use of port 465 and makes the guidance in this document less consistent.
- o Discussion of pinning certificates is new and may be inadequate. Suggestions to improve the text are welcome.
- o This document should reference draft-ietf-uta-tls-bcp and possibly other guidance documents. Suggested text on where/how to reference this and possibly other TLS guidance (e.g., must staple). would be welcome.
- o One author believes that the security latch model is complementary with draft-ietf-dane-smtp-with-dane-02 but hasn't thought about the issues in depth. We welcome feedback on this point.
- o The three involved authors are willing to merge draft-melnikov-email-tls-certs into this document. However, this will take time so we are only willing to do so if there is rough consensus on the decision (so it's a one time action) and doing so will not significantly delay publication.
- o It might make sense to split this in two or more documents if it's getting too long to evaluate in one IETF last call. In particular, it might make sense to put implementation requirements and service provider requirements in separate documents. The authors prefer to edit one document for now and defer discussion of splitting the document until all technical issues are resolved.
- o The use of SRV records [RFC6186] for account setup or refresh is presently not secure from DNS active attacks unless DNSSEC is used. As this document is now focusing on MUA security/privacy, discussing how to do SRV record account setup or account refresh securely, probably using DANE, would be in scope for this document. It has been suggested that we add this.

- o This document does not cover use of TLS with SMTP relay.

Appendix C. Change Log

Changes since -01:

- o Updated abstract, introduction and document structure to focus more on mail user agent privacy assurance.
- o Added email account privacy section, also moving section on account setup using SRV records to that section.
- o Finished writing IANA considerations section
- o Remove provisional concept and instead have server explicitly list security tags clients should latch.
- o Added note that rules for the submissions port follow the same rules as those for the submit port.
- o Reference and update advice in [RFC5068].
- o Fixed typo in Client Certificate Authentication section.
- o Removed tls-pfs security latch and all mention of perfect forward secrecy as it was controversial.
- o Added reference to HSTS.

Changes since -00:

- o Rewrote introduction to merge ideas from draft-moore-email-tls-00.
- o Added Implicit TLS section, Account configuration section and IANA port registration updates based on draft-moore-email-tls-00.
- o Add protocol details necessary to standardize implicit TLS for POP/IMAP/submission, using ideas from draft-melnikov-pop3-over-tls.
- o Reduce initial set of security tags based on feedback.
- o Add deep status concept to allow a window for software updates to be backed out before latches make that problematic, as well as to provide service providers with a mechanism they can use to assist customers in the event of a privacy failure.

- o Add DNS SRV section from draft-moore-email-tls-00.
- o Write most of the missing IANA considerations section.
- o Rewrite most of implementation requirements section based more on draft-moore-email-tls-00. Remove new cipher requirements for now because those may be dealt with elsewhere.

Appendix D. Acknowledgements

Many thanks to Ned Freed for discussion of the initial latch concepts in this document. Thanks to Alexey Melnikov for draft-melnikov-pop3-over-tls-02, which was the basis of the POP3 implicit TLS text. Thanks to Dan Newman and Alexey Melnikov for review feedback. Thanks to Paul Hoffman for interesting feedback in initial conversations about this idea.

Authors' Addresses

Keith Moore
Network Heretics
PO Box 1934
Knoxville, TN 37901
US

Email: moore@network-heretics.com

Chris Newman
Oracle
440 E. Huntington Dr., Suite 400
Arcadia, CA 91006
US

Email: chris.newman@oracle.com

Internet Engineering Task Force
Internet-Draft
Updates: 5246,4346,2246 (if approved)
Intended status: Standards Track
Expires: October 13, 2014

A. Popov
Microsoft Corp.
April 11, 2014

Prohibiting RC4 Cipher Suites
draft-popov-tls-prohibiting-rc4-02

Abstract

This document requires that Transport Layer Security (TLS) clients and servers never negotiate the use of RC4 cipher suites when they establish connections.

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 13, 2014.

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
1.1. Requirements Language	2
2. Changes to TLS	2
3. Acknowledgements	3
4. IANA Considerations	3
5. Security Considerations	3
6. References	3
6.1. Normative References	3
6.2. Informative References	3
Appendix A. RC4 Cipher Suites	4
Author's Address	4

1. Introduction

RC4 is a stream cipher described in [SCH], which is widely supported, and often preferred, by TLS servers. However, RC4 has long been known to have a variety of cryptographic weaknesses, e.g. [PAU], [MAN], [FLU]. Recent cryptanalysis results [ALF] exploit biases in the RC4 keystream to recover repeatedly encrypted plaintexts.

These recent results are on the verge of becoming practically exploitable; currently they require 2^{26} sessions or 13×2^{30} encryptions. As a result, RC4 can no longer be seen as providing a sufficient level of security for TLS sessions.

This document requires that TLS ([RFC5246], [RFC4346], [RFC2246]) clients and servers never negotiate the use of RC4 cipher suites.

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 [RFC2119].

2. Changes to TLS

Because of the deficiencies noted in Section 1:

- o TLS clients MUST NOT include RC4 cipher suites in the ClientHello message.
- o TLS servers MUST NOT select an RC4 cipher suite when a TLS client sends such a cipher suite in the ClientHello message.

- o If the TLS client only offers RC4 cipher suites, the TLS server MUST terminate the handshake. The TLS server MAY send the `insufficient_security` fatal alert in this case.

Appendix A lists the RC4 cipher suites defined for TLS.

3. Acknowledgements

This document was inspired by discussions with Magnus Nystrom, Eric Rescorla, Joseph Salowey, Yaron Sheffer, Nagendra Modadugu and others on the TLS mailing list.

4. IANA Considerations

This memo includes no request to IANA.

5. Security Considerations

This document helps maintain the security guarantees of the TLS protocol by prohibiting the use of the RC4-based cipher suites (listed in Appendix A), which do not provide a sufficiently high level of security.

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.
- [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.

6.2. Informative References

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

- [FLU] Fluhrer, S., Mantin, I., and A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4. Selected Areas in Cryptography, pp. 1-24", 2001.
- [MAN] Mantin, I. and A. Shamir, "A Practical Attack on Broadcast RC4. FSE, pp. 152-164.", 2001.
- [PAU] Paul, G. and S. Maitra, "Permutation after RC4 Key Scheduling Reveals the Secret Key. In Proceedings of the 14th Workshop on Selected Areas in Cryptography (SAC), pp. 360-377, vol. 4876, LNCS, Springer.", 2007.
- [SCH] Schneier, B., "Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd ed.", 1996.

Appendix A. RC4 Cipher Suites

The following cipher suites defined for TLS use RC4:

- o TLS_RSA_WITH_RC4_128_MD5
- o TLS_RSA_WITH_RC4_128_SHA
- o TLS_DH_anon_WITH_RC4_128_MD5
- o TLS_RSA_EXPORT_WITH_RC4_40_MD5
- o TLS_DH_anon_EXPORT_WITH_RC4_40_MD5

Author's Address

Andrei Popov
Microsoft Corp.
One Microsoft Way
Redmond, WA 98052
USA

Email: andreipo@microsoft.com

Network Working Group
Internet-Draft
Updates: 6120 (if approved)
Intended status: Standards Track
Expires: September 5, 2014

P. Saint-Andre
&yet
T. Alkemade
March 4, 2014

Use of Transport Layer Security (TLS) in the Extensible Messaging and
Presence Protocol (XMPP)
draft-saintandre-xmpp-tls-06

Abstract

This document provides recommendations for the use of Transport Layer Security (TLS) in the Extensible Messaging and Presence Protocol (XMPP). This document updates RFC 6120.

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 September 5, 2014.

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. Terminology	2
3. Discussion Venue	3
4. Recommendations	3
4.1. Support for TLS	3
4.2. Protocol Versions	3
4.3. Cipher Suites	3
4.4. Public Key Length	3
4.5. Compression	3
4.6. Session Resumption	4
4.7. Authenticated Connections	4
4.8. Unauthenticated Connections	4
4.9. Server Name Indication	4
4.10. Human Factors	5
5. Implementation Notes	5
6. IANA Considerations	5
7. Security Considerations	5
8. References	6
8.1. Normative References	6
8.2. Informative References	6
8.3. URIs	7
Appendix A. Acknowledgements	8
Authors' Addresses	8

1. Introduction

The Extensible Messaging and Presence Protocol (XMPP) [RFC6120] (along with its precursor, the so-called "Jabber protocol") has used Transport Layer Security (TLS) [RFC5246] (along with its precursor, Secure Sockets Layer or SSL) since 1999. Both [RFC6120] and its predecessor [RFC3920] provided recommendations regarding the use of TLS in XMPP. In order to address the evolving threat model on the Internet today (see, for example, [I-D.trammell-perpass-ppa]), this document provides stronger recommendations (see also [I-D.sheffer-tls-bcp]). This document updates [RFC6120].

2. Terminology

Various security-related terms are to be understood in the sense defined in [RFC4949].

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].

3. Discussion Venue

The discussion venue for this document is the mailing list of the XMPP Working Group, for which archives and subscription information can be found at [1]. Discussion might also occur on the mailing list of the UTA Working Group, for which archives and subscription information can be found at [2].

4. Recommendations

4.1. Support for TLS

Support for TLS (specifically, the XMPP profile of STARTTLS) is mandatory for XMPP implementations, as already specified in [RFC6120] and its predecessor [RFC3920].

If the server to which an XMPP client or peer server connects does not offer a stream feature of `<starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>` (thus indicating that it is an XMPP 1.0 server that supports TLS), the initiating entity MUST NOT proceed with the stream negotiation and MUST instead abort the connection attempt. Although XMPP servers SHOULD include the `<required/>` child element to indicate that negotiation of TLS is mandatory, clients and peer servers MUST NOT depend on receiving the `<required/>` flag in determining whether TLS will be enforced for the stream.

4.2. Protocol Versions

Implementations MUST follow the recommendations in [I-D.sheffer-tls-bcp] as to supporting various TLS versions and avoiding fallback to SSL.

4.3. Cipher Suites

Implementations MUST follow the recommendations in [I-D.sheffer-tls-bcp].

4.4. Public Key Length

Implementations MUST follow the recommendations in [I-D.sheffer-tls-bcp].

4.5. Compression

Implementations MUST follow the recommendations in [I-D.sheffer-tls-bcp].

XMPP supports an application-layer compression technology [XEP-0138], which might have slightly stronger security properties than TLS (at least because it is enabled after SASL authentication, as described in [XEP-0170]).

4.6. Session Resumption

Implementations MUST follow the recommendations in [I-D.sheffer-tls-bcp].

Use of session IDs [RFC5246] is RECOMMENDED instead of session tickets [RFC5077], since XMPP does not in general use state management technologies such as tickets or "cookies" [RFC6265].

Note that, in XMPP, TLS session resumption can be used in concert with the XMPP Stream Management extension; see [XEP-0198] for further details.

4.7. Authenticated Connections

Both the core XMPP specification [RFC6120] and the "CertID" specification [RFC6125] provide recommendations and requirements for certificate validation in the context of authenticated connections. This document does not supersede those specifications. Wherever possible, it is best to prefer authenticated connections (along with SASL [RFC4422]), as already stated in the core XMPP specification [RFC6120]. In particular, clients MUST authenticate servers.

4.8. Unauthenticated Connections

Given the pervasiveness of passive eavesdropping, even an unauthenticated connection might be better than an unencrypted connection (this is similar to the "better than nothing security" approach for IPsec [RFC5386]). In particular, because of current deployment challenges for authenticated connections between XMPP servers (see [I-D.ietf-xmpp-dna] for details), it might be reasonable for XMPP server implementations to accept unauthenticated connections when the Server Dialback protocol [XEP-0220] is used for weak identity verification; this will at least enable encryption of server-to-server connections. Unauthenticated connections include connections negotiated using anonymous Diffie-Hellman algorithms or using self-signed certificates, among other scenarios.

4.9. Server Name Indication

Although there is no harm in supporting the TLS Server Name Indication (SNI) extension [RFC6066], this is not necessary since the

same function is served in XMPP by the 'to' address of the initial stream header as explained in Section 4.7.2 of [RFC6120].

4.10. Human Factors

It is RECOMMENDED that XMPP clients provide ways for end users (and that XMPP servers provide ways for administrators) to complete the following tasks:

- o Determine if a client-to-server or server-to-server connection is encrypted and authenticated.
- o Determine the version of TLS used for a client-to-server or server-to-server connection.
- o Inspect the certificate offered by an XMPP server.
- o Determine the cipher suite used to encrypt a connection.
- o Be warned if the certificate changes for a given server.

5. Implementation Notes

Some governments enforce legislation prohibiting the export of strong cryptographic technologies. Nothing in this document ought to be taken as advice to violate such prohibitions.

6. IANA Considerations

This document requests no actions of the IANA.

7. Security Considerations

As noted in "A Threat Model for Pervasive Passive Surveillance" [I-D.trammell-perpass-ppa], the use of TLS can help limit the information available for correlation to the network and transport layer headers as opposed to the application layer. As typically deployed, XMPP technologies do not leave application-layer routing data (such as XMPP 'to' and 'from' addresses) at rest on intermediate systems, since there is only one hop between any two given XMPP servers. As a result, encrypting all hops (sending client to sender's server, sender's server to recipient's server, recipient's server to recipient's client) can help to limit the amount of "metadata" that might leak.

It is possible that XMPP servers themselves might be compromised. In that case, per-hop encryption would not protect XMPP communications, and even end-to-end encryption of (parts of) XMPP stanza payloads

would leave addressing information and XMPP roster data in the clear. By the same token, it is possible that XMPP clients (or the end-user devices on which such clients are installed) could also be compromised, leaving users utterly at the mercy of an adversary.

This document, along with actions currently being taken to strengthen the security of the XMPP network, do not assume widespread compromise of XMPP servers and clients or their underlying operating systems or hardware. Thus it is assumed that ubiquitous use of per-hop TLS channel encryption and more significant deployment of end-to-end object encryption technologies will serve to protect XMPP communications to a measurable degree, compared to the alternatives.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", RFC 4949, August 2007.
- [RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", RFC 5077, January 2008.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.
- [RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011.
- [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.

8.2. Informative References

- [I-D.ietf-xmpp-dna] Saint-Andre, P. and M. Miller, "Domain Name Associations (DNA) in the Extensible Messaging and Presence Protocol (XMPP)", draft-ietf-xmpp-dna-05 (work in progress), February 2014.

- [I-D.sheffer-tls-bcp]
Sheffer, Y., Holz, R., and P. Saint-Andre,
"Recommendations for Secure Use of TLS and DTLS", draft-
sheffer-tls-bcp-02 (work in progress), February 2014.
- [I-D.trammell-perpass-ppa]
Trammell, B., Borkmann, D., and C. Huitema, "A Threat
Model for Pervasive Passive Surveillance", draft-trammell-
perpass-ppa-01 (work in progress), November 2013.
- [RFC3920] Saint-Andre, P., Ed., "Extensible Messaging and Presence
Protocol (XMPP): Core", RFC 3920, October 2004.
- [RFC4422] Melnikov, A. and K. Zeilenga, "Simple Authentication and
Security Layer (SASL)", RFC 4422, June 2006.
- [RFC5386] Williams, N. and M. Richardson, "Better-Than-Nothing
Security: An Unauthenticated Mode of IPsec", RFC 5386,
November 2008.
- [RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions:
Extension Definitions", RFC 6066, January 2011.
- [RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265,
April 2011.
- [XEP-0138]
Hildebrand, J. and P. Saint-Andre, "Stream Compression",
XSF XEP 0138, May 2009.
- [XEP-0170]
Saint-Andre, P., "Recommended Order of Stream Feature
Negotiation", XSF XEP 0170, January 2007.
- [XEP-0198]
Karneges, J., Saint-Andre, P., Hildebrand, J., Forno, F.,
Cridland, D., and M. Wild, "Stream Management", XSF XEP
0198, June 2011.
- [XEP-0220]
Miller, J., Saint-Andre, P., and P. Hancke, "Server
Dialback", XSF XEP 0220, September 2013.

8.3. URIs

- [1] <https://www.ietf.org/mailman/listinfo/xmpp>
- [2] <https://www.ietf.org/mailman/listinfo/uta>

Appendix A. Acknowledgements

Thanks to the following individuals for their input: Dave Cridland, Philipp Hancke, Olle Johansson, Steve Kille, Tobias Markmann, Matt Miller, and Rene Treffer.

Authors' Addresses

Peter Saint-Andre
&yet

Email: ietf@stpeter.im

Thijs Alkemade

Email: me@thijsalkema.de

UTA
Internet-Draft
Intended status: Best Current Practice
Expires: August 17, 2014

Y. Sheffer
Porticor
R. Holz
TUM
P. Saint-Andre
&yet
February 13, 2014

Recommendations for Secure Use of TLS and DTLS
draft-sheffer-tls-bcp-02

Abstract

Transport Layer Security (TLS) and Datagram Transport Security Layer (DTLS) are widely used to protect data exchanged over application protocols such as HTTP, SMTP, IMAP, POP, SIP, and XMPP. Over the last few years, several serious attacks on TLS have emerged, including attacks on its most commonly used cipher suites and modes of operation. This document provides recommendations for improving the security of both software implementations and deployed services that use TLS and DTLS.

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 August 17, 2014.

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	3
3. Recommendations	3
3.1. Protocol Versions	3
3.2. Fallback to SSL	4
3.3. Cipher Suites	4
3.4. Public Key Length	6
3.5. Compression	6
3.6. Session Resumption	6
4. Detailed Guidelines	6
4.1. Cipher Suite Negotiation Details	7
4.2. Alternative Cipher Suites	7
5. IANA Considerations	8
6. Security Considerations	8
6.1. AES-GCM	8
6.2. Forward Secrecy	8
7. Acknowledgements	9
8. References	9
8.1. Normative References	9
8.2. Informative References	10
Appendix A. Appendix: Change Log	11
A.1. -02	11
A.2. -01	11
A.3. -00	12
Authors' Addresses	12

1. Introduction

Transport Layer Security (TLS) and Datagram Transport Security Layer (DTLS) are widely used to protect data exchanged over application protocols such as HTTP, SMTP, IMAP, POP, SIP, and XMPP. Over the last few years, several serious attacks on TLS have emerged, including attacks on its most commonly used cipher suites and modes of operation. For instance, both AES-CBC and RC4, which together comprise most current usage, have been attacked in the context of TLS. A companion document [I-D.sheffer-uta-tls-attacks] provides detailed information about these attacks.

Because of these attacks, those who implement and deploy TLS and DTLS need updated guidance on how TLS can be used securely. Note that this document provides guidance for deployed services, as well as software implementations. In fact, this document calls for the deployment of algorithms that are widely implemented but not yet widely deployed.

The recommendations herein take into consideration the security of various mechanisms, their technical maturity and interoperability, and their prevalence in implementations at the time of writing. These recommendations apply 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.

Community knowledge about the strength of various algorithms and feasible attacks can change quickly, and experience shows that a security BCP is a point-in-time statement. Readers are advised to seek out any errata or updates that apply to this document.

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. Recommendations

3.1. Protocol Versions

It is important both to stop using old, less secure versions of SSL/TLS and to start using modern, more secure versions. Therefore:

- o Implementations MUST NOT negotiate SSL version 2.

Rationale: SSLv2 has serious security vulnerabilities [RFC6176].

- o Implementations SHOULD NOT negotiate SSL version 3.

Rationale: SSLv3 [RFC6101] was an improvement over SSLv2 and plugged some significant security holes, but did not support strong cipher suites.

- o Implementations MAY negotiate TLS version 1.0 [RFC2246].

Rationale: TLS 1.0 (published in 1999) includes a way to downgrade the connection to SSLv3 and does not support more modern, strong cipher suites.

- o Implementations MAY negotiate TLS version 1.1 [RFC4346].

Rationale: TLS 1.1 (published in 2006) prevents downgrade attacks to SSL, but does not support certain stronger cipher suites.

- o Implementations MUST support, and prefer to negotiate, TLS version 1.2 [RFC5246].

Rationale: Several stronger cipher suites are available only with TLS 1.2 (published in 2008).

As of the date of this writing, the latest version of TLS is 1.2. When TLS is updated to a newer version, this document will be updated to recommend support for the latest version. If this document is not updated in a timely manner, it can be assumed that support for the latest version of TLS is recommended.

3.2. Fallback to SSL

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 only about 3% of the current web server population. Therefore, by default clients SHOULD NOT fall back from TLS to SSLv3.

3.3. Cipher Suites

It is important both to stop using old, insecure cipher suites and to start using modern, more secure cipher suites. Therefore:

- o Implementations MUST NOT negotiate the NULL cipher suites.

Rationale: The NULL cipher suites offer no encryption whatsoever and thus are completely insecure.

- o Implementations MUST NOT negotiate RC4 cipher suites

Rationale: The RC4 stream cipher has a variety of cryptographic weaknesses, as documented in [I-D.popov-tls-prohibiting-rc4].

- o Implementations MUST NOT negotiate cipher suites offering only so-called "export-level" encryption (including algorithms with 40 bits or 56 bits of security).

Rationale: These cipher suites are deliberately "dumbed down" and are very easy to break.

- o Implementations SHOULD NOT negotiate cipher suites that use algorithms offering less than 128 bits of security (even if they advertise more bits, such as the 168-bit 3DES cipher suites).

Rationale: Although these cipher suites are not actively subject to breakage, their useful life is short enough that stronger cipher suites are desirable.

- o Implementations SHOULD prefer cipher suites that use algorithms with at least 128 (and, if possible, 256) bits of security.

Rationale: Although the useful life of such cipher suites is unknown, it is probably at least several years for the 128-bit ciphers and "until the next fundamental technology breakthrough" for 256-bit ciphers.

- o Implementations MUST support, and SHOULD prefer to negotiate, cipher suites offering forward secrecy, such as those in the "EDH", "DHE", and "ECDHE" families.

Rationale: Forward secrecy (sometimes called "perfect forward secrecy") prevents the recovery of information that was encrypted with older session keys, thus limiting the amount of time during which attacks can be successful.

Given the foregoing considerations, implementation of the following cipher suites is RECOMMENDED (see [RFC5289] for details):

- o TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- o TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- o TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- o TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

We suggest that TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 be preferred in general.

Unfortunately, those cipher suites are supported only in TLS 1.2 since they are authenticated encryption (AEAD) algorithms [RFC5116]. A future version of this document might recommend cipher suites for earlier versions of TLS.

[RFC4492] allows clients and servers to negotiate ECDH parameters (curves). Clients and servers SHOULD prefer verifiably random curves (specifically Brainpool P-256, brainpoolp256r1 [RFC7027]), and fall back to the commonly used NIST P-256 (secp256r1) curve [RFC4492]. In

addition, clients SHOULD send an `ec_point_formats` extension with a single element, "uncompressed".

3.4. Public Key Length

Because Diffie-Hellman keys of 1024 bits are estimated to be roughly equivalent to 80-bit symmetric keys, it is better to use longer keys for the "DH" family of cipher suites. Unfortunately, some existing software cannot handle (or cannot easily handle) key lengths greater than 1024 bits. The most common workaround for these systems is to prefer the "ECDHE" family of cipher suites instead of the "DH" family, then use longer keys. Key lengths of at least 2048 bits are RECOMMENDED, since they are estimated to be roughly equivalent to 112-bit symmetric keys and might be sufficient for at least the next 10 years. In addition to 2048-bit server certificates, the use of SHA-256 fingerprints is RECOMMENDED (see [CAB-Baseline] for more details).

Note: The foregoing recommendations are preliminary and will likely be corrected and enhanced in a future version of this document.

3.5. Compression

Implementations and deployments SHOULD disable TLS-level compression ([RFC5246], Sec. 6.2.2).

3.6. Session Resumption

If TLS session resumption is used, care ought to be taken to do so safely. In particular, the resumption information (either session IDs [RFC5246] or session tickets [RFC5077]) needs to be authenticated and encrypted to prevent modification or eavesdropping by an attacker. For session tickets, a strong cipher suite SHOULD be used when encrypting the ticket (as least as strong as the main TLS cipher suite); ticket keys MUST be changed regularly, e.g. once every week, so as not to negate the effect of forward secrecy. Session ticket validity SHOULD be limited to a reasonable duration (e.g. 1 day), so as not to negate the benefits of forward secrecy.

4. Detailed Guidelines

The following sections provide more detailed information about the recommendations listed above.

4.1. Cipher Suite Negotiation Details

Clients SHOULD include `TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256` 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.

Servers SHOULD 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 compelling reasons (e.g., seriously degraded 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`.

4.2. Alternative Cipher Suites

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:

- o There are no protocol mechanisms to negotiate the DH groups or parameter lengths supported by client and server.
- o 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 of 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 [RFC5289]
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. IANA Considerations

This document requests no actions of IANA.

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. Forward Secrecy

Forward secrecy (also often called Perfect Forward Secrecy or "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:

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

- o 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.

7. 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.

8. References

8.1. 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.

- [RFC5289] Rescorla, E., "TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)", RFC 5289, August 2008.
- [RFC6176] Turner, S. and T. Polk, "Prohibiting Secure Sockets Layer (SSL) Version 2.0", RFC 6176, March 2011.
- [RFC7027] Merkle, J. and M. Lochter, "Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS)", RFC 7027, October 2013.

8.2. Informative References

- [CAB-Baseline]
"Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates Version 1.1.6", 2013,
<<https://www.cabforum.org/documents.html>>.
- [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.
- [I-D.popov-tls-prohibiting-rc4]
Popov, A., "Prohibiting RC4 Cipher Suites", draft-popov-tls-prohibiting-rc4-01 (work in progress), October 2013.
- [I-D.sheffer-uta-tls-attacks]
Sheffer, Y., Holz, R., and P. Saint-Andre, "Summarizing Current Attacks on TLS and DTLS", draft-sheffer-uta-tls-attacks-00 (work in progress), February 2014.
- [Kleinjung2010]
Kleinjung, T., "Factorization of a 768-Bit RSA Modulus", CRYPTO 10, 2010.
- [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.
- [RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", RFC 5077, January 2008.

- [RFC5116] McGrew, D., "An Interface and Algorithms for Authenticated Encryption", RFC 5116, January 2008.
- [RFC6101] Freier, A., Karlton, P., and P. Kocher, "The Secure Sockets Layer (SSL) Protocol Version 3.0", RFC 6101, August 2011.
- [RFC6460] Salter, M. and R. Housley, "Suite B Profile for Transport Layer Security (TLS)", RFC 6460, January 2012.
- [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. -02

- o Reorganized the content to focus on recommendations.
- o Moved description of attacks to a separate document (draft-sheffer-uta-tls-attacks).
- o Strengthened recommendations regarding session resumption.

A.2. -01

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

A.3. -00

- o 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

Peter Saint-Andre
&yet

Email: ietf@stpeter.im

uta
Internet-Draft
Intended status: Informational
Expires: August 11, 2014

Y. Sheffer
Porticor
R. Holz
TUM
P. Saint-Andre
&yet
February 7, 2014

Summarizing Current Attacks on TLS and DTLS
draft-sheffer-uta-tls-attacks-00

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 summarizes these attacks, with the goal of motivating generic and protocol-specific recommendations on the usage of TLS and DTLS.

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 August 11, 2014.

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	3
1.1.	Conventions used in this document	3
2.	Attacks on TLS	3
2.1.	BEAST	3
2.2.	Lucky Thirteen	3
2.3.	Attacks on RC4	4
2.4.	Compression Attacks: CRIME and BREACH	4
3.	Security Considerations	4
4.	IANA Considerations	4
5.	Acknowledgements	4
6.	References	5
6.1.	Normative References	5
6.2.	Informative References	5
Appendix A.	Appendix: Change Log	6
A.1.	-00	6
	Authors' Addresses	6

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.

This situation motivated the creation of the UTA working group, which is tasked with the creation of generic and protocol-specific recommendation for the use of TLS and DTLS.

"Attacks always get better; they never get worse" (ironically, this saying is attributed to the NSA). This list of attacks describes our knowledge as of this writing. It seems likely that new attacks will be invented in the future.

For a more detailed discussion of the attacks listed here, the interested reader is referred to [Attacks-iSec].

1.1. 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].

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, the predictable initialization vector) 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 TIME attack [TIME] and the later BREACH attack [BREACH] both make similar use of HTTP-level compression to decrypt secret data passed in the HTTP response. We note that compression of the HTTP message body is much more prevalent than compression at the TLS level.

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 [TBD] are adopted.

3. Security Considerations

This document describes protocol attacks in an informational manner, and in itself does not have any security implications. Its companion documents certainly do.

4. IANA Considerations

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

This document requires no IANA actions.

5. Acknowledgements

We would like to thank Stephen Farrell, Simon Josefsson, Yoav Nir,

Kenny Paterson, Patrick Pelletier, and Rich Salz for their review of a previous version of this document.

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

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

6.2. Informative References

- [I-D.popov-tls-prohibiting-rc4]
Popov, A., "Prohibiting RC4 Cipher Suites",
draft-popov-tls-prohibiting-rc4-01 (work in progress),
October 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/>>.
- [TIME] Be'ery, T. and A. Shulman, "A Perfect CRIME? Only TIME Will Tell", Black Hat Europe 2013, 2013, <<https://media.blackhat.com/eu-13/briefings/Beery/bh-eu-13-a-perfect-crime-beery-wp.pdf>>.
- [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>>.

[Attacks-iSec]

Sarkar, P. and S. Fitzgerald, "Attacks on SSL, a comprehensive study of BEAST, CRIME, TIME, BREACH, Lucky13 and RC4 biases", 8 2013, <https://www.isecpartners.com/media/106031/ssl_attacks_survey.pdf>.

[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>>.

Appendix A. Appendix: Change Log

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

A.1. -00

- o Initial version, extracted from draft-sheffer-tls-bcp-01.

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

Peter Saint-Andre
&yet

Email: ietf@stpeter.im

