

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: August 17, 2017

J. Fenton
Altmode Networks
February 13, 2017

SMTP Require TLS Option
draft-fenton-smtp-require-tls-03

Abstract

The SMTP STARTTLS option, used in negotiating transport-level encryption of SMTP connections, is not as useful from a security standpoint as it might be because of its opportunistic nature; message delivery is prioritized over security. This document describes a complementary SMTP service extension, REQUIRETLS. If the REQUIRETLS option is used when sending a message, it asserts a request on the part of the message sender to override the default negotiation of TLS, either by requiring that TLS be negotiated when the message is relayed, or by requesting that policy mechanisms such as SMTP STS and DANE be ignored when relaying a high priority message.

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, 2017.

Copyright Notice

Copyright (c) 2017 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	3
2. The REQUIRETLS Service Extension	3
3. REQUIRETLS Semantics	5
3.1. REQUIRETLS Receipt Requirements	5
3.2. REQUIRETLS Sender Requirements	5
3.2.1. Sending with TLS Required	5
3.2.2. Sending with TLS Optional	6
3.3. REQUIRETLS Submission	7
3.4. Delivery of REQUIRETLS messages	7
4. Non-delivery message handling	7
5. Mailing list considerations	8
6. IANA Considerations	8
7. Security Considerations	8
7.1. Passive attacks	9
7.2. Active attacks	9
7.3. Bad Actor MTAs	9
8. Acknowledgements	10
9. Revision History	10
9.1. Changes Since -02 Draft	10
9.2. Changes Since -01 Draft	10
9.3. Changes Since -00 Draft	10
10. References	11
10.1. Normative References	11
10.2. Informative References	12
Author's Address	13

1. Introduction

The SMTP [RFC5321] STARTTLS service extension [RFC3207] provides a means by which an SMTP server and client can establish a Transport Layer Security (TLS) protected session for the transmission of email messages. By default, TLS is used only upon mutual agreement (successful negotiation) of STARTTLS between the client and server; if this is not possible, the message is sent without transport encryption. Furthermore, it is common practice for the client to negotiate TLS even if the SMTP server's certificate fails to authenticate it.

Policy mechanisms such as DANE [RFC7672] and SMTP STS [I-D.ietf-uta-mta-sts] may impose requirements for the use of TLS for email destined for some domains. However, such policies do not allow the sender to specify which messages are more sensitive and require transport-level encryption, and which ones are urgent and ought to be relayed even if TLS cannot be negotiated successfully.

The default opportunistic nature of SMTP TLS enables several "on the wire" attacks on SMTP security between MTAs. These include passive eavesdropping on connections for which TLS is not used, interference in the SMTP protocol to prevent TLS from being negotiated (presumably followed by eavesdropping), and insertion of a man-in-the-middle attacker taking advantage of the lack of server authentication by the client. Attacks are more described in more detail in the Security Considerations section of this document.

The REQUIRETLS SMTP service extension allows the SMTP client to specify that a given message sent during a particular session **MUST** be sent over a TLS protected session with specified security characteristics, or conversely that delivery should be prioritized over ability to negotiate TLS. For messages requiring TLS negotiation, it also requires that the SMTP server advertise that it also supports REQUIRETLS, in effect promising that it will honor the requirement to require TLS transmission and REQUIRETLS support for onward transmission of messages specifying that requirement.

1.1. Requirements Language

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

2. The REQUIRETLS Service Extension

1. The textual name of the extension is "Require TLS".
2. The EHLO keyword value associated with this extension is "REQUIRETLS".
3. One MAIL FROM option is defined by this extension.
4. Two new SMTP status codes are defined by this extension to convey error conditions resulting from failure of the client to negotiate a TLS connection with the required security and as a result of an attempt to send to a server not also supporting the REQUIRETLS extension.

In order to specify REQUIRETLS treatment for a given message, the REQUIRETLS option is specified on the MAIL FROM command when that message is transmitted. With the exception of REQUIRETLS=NO (described below), this option MUST only be specified in the context of an SMTP session meeting the security requirements that have been specified:

- o The session itself MUST employ TLS transmission, unless the NO parameter is specified.
- o Any server authentication requirements specified as an option to the REQUIRETLS option (see below) MUST have been satisfied in establishing the current session.

An optional parameter to the REQUIRETLS MAIL FROM option specifies the requirements for server authentication that MUST be used for any onward transmission of the following message. The parameter takes the form of either a single value or comma-separated list, separated from the REQUIRETLS option by a single "=" (equals-sign) character. If present, the parameter MUST take one or more of the following values:

- o CHAIN - The certificate presented by the SMTP server MUST verify successfully in a trust chain leading to a certificate trusted by the SMTP client. The choice of trusted (root) certificates by the client is at their own discretion. The client MAY choose to use the certificate set maintained by the CA/B forum [citation needed] for this purpose.
- o DANE - The certificate presented by the SMTP server MUST verify successfully using DANE as specified in RFC 7672 [RFC7672].
- o DNSSEC - The server MUST confirm that any MX record or CNAME lookup used to locate the SMTP server must be DNSSEC [RFC4035] signed and valid.
- o NO - The SMTP client SHOULD attempt to send the message regardless of its ability to negotiate STARTTLS with the SMTP server, ignoring policy-based mechanisms, if any, asserted by the recipient domain. Nevertheless, the client MAY negotiate STARTTLS with the server if available. If the NO parameter is present, any other REQUIRETLS parameter MUST NOT be used.

The CHAIN and DANE parameters are additive; if both are specified, either method of certificate validation is acceptable. If neither CHAIN nor DANE is specified, the certificate presented by the SMTP server is not required to be verified.

3. REQUIRETLS Semantics

3.1. REQUIRETLS Receipt Requirements

Upon receipt of the REQUIRETLS option on a MAIL FROM command during the receipt of a message, an SMTP server MUST tag that message as needing REQUIRETLS handling with the specified option(s). The manner in which this tagging takes place is implementation-dependent. If the message is being locally aliased and redistributed to multiple addresses, all instances of the message MUST be tagged in the same manner.

3.2. REQUIRETLS Sender Requirements

3.2.1. Sending with TLS Required

When sending a message tagged with REQUIRETLS other than the REQUIRETLS=NO option, the sending (client) MTA MUST:

1. Look up the SMTP server to which the message is to be sent. If the DNSSEC option is included in the message tag, the MX record lookups in this process MUST use DNSSEC verification and the response(s) MUST be DNSSEC-signed in order to ensure the integrity of the resource identifier [RFC6125] used to authenticate the SMTP server.
2. Open an SMTP session with the peer SMTP server using the EHLO verb. The server MUST advertise the REQUIRETLS capability.
3. Establish a TLS-protected SMTP session with its peer SMTP server and authenticate the server's certificate with the specified authentication method.
4. The SMTP client SHOULD also require that meaningfully secure cipher algorithms and key lengths be negotiated with the server. The choices of key lengths and algorithms change over time, so a specific requirement is not presented here.

If any of the above steps fail, the client SHOULD issue a QUIT to the server and repeat steps 2-4 with each host on the recipient domain's list of MX hosts in an attempt to find a mail path that meets the sender's requirements. If there are no more MX hosts or if the MX record lookup is not DNSSEC-protected and DNSSEC verification is required, the client MUST NOT transmit the message and MUST issue an SMTP QUIT command to the server. The client MAY send other, unprotected, messages to that server prior to issuing the QUIT if it has any.

Following such a failure, the SMTP client MUST send a non-delivery notification to the reverse-path of the failed message as described in section 3.6 of [RFC5321]. The following status codes [RFC5248] SHOULD be used:

- o DNSSEC lookup failure: 5.x.x DNSSEC lookup required
- o REQUIRETLS not supported by server: 5.7.x REQUIRETLS needed
- o Unable to establish TLS-protected SMTP session: 5.7.10 Encryption needed

Refer to Section 4. for further requirements regarding non-delivery messages.

If all REQUIRETLS requirements have been met, transmit the message, issuing the REQUIRETLS option on the MAIL FROM command with the required option(s), if any.

3.2.2. Sending with TLS Optional

Messages tagged REQUIRETLS=NO are handled differently from other REQUIRETLS messages, as follows. When sending a message tagged with REQUIRETLS=NO, the sending (client) MTA MUST:

- o Look up the SMTP server to which the message is to be sent as described in [RFC5321].
- o Open an SMTP session with the peer SMTP server using the EHLO verb. Attempt to negotiate STARTTLS if possible, and follow any policy published by the recipient domain, but do not fail if this is unsuccessful.
- o If the server does not advertise the REQUIRETLS capability, send the message in the usual manner (without the REQUIRETLS option, because the server will not understand the option).
- o If the server advertises the REQUIRETLS capability, send the message with the REQUIRETLS=NO option.

Some SMTP servers that are configured to expect STARTTLS connections as a matter of policy may not accept messages in the absence of STARTTLS. This MUST be expected, and a non-delivery notification returned to the sender.

Messages tagged with the REQUIRETLS=NO option will be sent without the option to SMTP servers not supporting REQUIRETLS. REQUIRETLS=NO MAY therefore not persist through multiple email relay hops.

3.3. REQUIRETLS Submission

An MUA or other agent making the initial introduction of a message to SMTP has authority to decide whether to require TLS, and if so, using what authentication method(s). It does so by issuing the REQUIRETLS option in the MAIL FROM command during message submission. This MAY be done based on a user interface selection, on a header field included in the message, or based on policy. The manner in which the decision to require TLS is made is implementation-dependent and is beyond the scope of this specification.

3.4. Delivery of REQUIRETLS messages

Messages are usually retrieved by end users using protocols other than SMTP such as IMAP [RFC3501], POP [RFC1939], or web mail systems. Mail delivery agents supporting REQUIRETLS SHOULD require that retrieval of messages requiring transport encryption take place over authenticated, encrypted channels.

4. Non-delivery message handling

Non-delivery ("bounce") messages usually contain important metadata about the message to which they refer, including the original message header. They therefore MUST be protected in the same manner as the original message. All non-delivery messages, whether resulting from a REQUIRETLS error or some other, MUST employ REQUIRETLS using the same authentication method(s) as the message that caused the error to occur.

It should be noted that the path from the origination of an error bounce message back to the MAIL FROM address may not share the same REQUIRETLS support as the forward path. Therefore, users of REQUIRETLS (other than REQUIRETLS=NO) are advised to make sure that they are capable of receiving mail using REQUIRETLS at the same authentication method(s) as messages they send. Otherwise, such non-delivery messages will be lost.

If unable to send a bounce message due to a REQUIRETLS failure (the return path not supporting the TLS requirements in the original message), the MTA sending the bounce message MAY send a redacted non-delivery message to the postmaster of the domain identified in the envelope-From address identifying the message only by Message-ID and indicating the type of failure. The original From, Return-path, To, Sender, Cc, and related header fields MUST NOT be included in this message.

Senders of messages specifying REQUIRETLS (other than REQUIRETLS=NO) are advised to consider the increased likelihood that bounce messages will be lost as a result of REQUIRETLS return path failure.

5. Mailing list considerations

Mailing lists, upon receipt of a message, originate new messages to list addresses, as distinct from an aliasing operation that redirects the original message, in some cases to multiple recipients. The requirement to preserve the REQUIRETLS tag and options therefore does not extend to mailing lists. REQUIRETLS users SHOULD use caution when sending to mailing lists and MUST NOT assume that REQUIRETLS applies to messages from the list operator to list members.

Mailing list operators MAY apply REQUIRETLS requirements in incoming messages to the resulting messages they originate. If this is done, they SHOULD also apply these requirements to administrative traffic, such as messages to moderators requesting approval of messages.

6. IANA Considerations

If published as an RFC, this draft requests the addition of the keyword REQUIRETLS to the SMTP Service Extensions Registry [MailParams].

If published as an RFC, this draft also requests the creation of a registry, REQUIRETLS Security Requirements, to be initially populated with the CHAIN, DANE, DNSSEC, and NO keywords.

If published as an RFC, this draft requests the addition of an entry to the Simple Mail Transfer Protocol (SMTP) Enhanced Status Codes Registry [SMTPStatusCodes] in the 5.7.YYY range to indicate lack of REQUIRETLS support by an SMTP server to which a message is being routed.

This section is to be removed during conversion into an RFC by the RFC Editor.

7. Security Considerations

The purpose of REQUIRETLS is to improve communications security for email by giving the originator of a message an expectation that it will be transmitted in an encrypted form "over the wire". When used, REQUIRETLS changes the traditional behavior of email transmission, which favors delivery over the ability to send email messages using transport-layer security, to one in which requested security takes precedence over delivery and domain-level policy.

The following considerations apply to STARTTLS other than the STARTTLS=NO option, since messages specifying that option are specifying less concern with transport security.

7.1. Passive attacks

REQUIRETLS is generally effective against passive attackers who are merely trying to eavesdrop on an SMTP exchange between an SMTP client and server. This assumes, of course, the cryptographic integrity of the TLS connection being used.

7.2. Active attacks

Active attacks against TLS encrypted SMTP connections can take many forms. One such attack is to interfere in the negotiation by changing the STARTTLS command to something illegal such as XXXXXXXX. This causes TLS negotiation to fail and messages to be sent in the clear, where they can be intercepted. REQUIRETLS detects the failure of STARTTLS and declines to send the message rather than send it insecurely.

A second form of attack is a man-in-the-middle attack where the attacker terminates the TLS connection rather than the intended SMTP server. This is possible when, as is commonly the case, the SMTP client either does not verify the server's certificate or establishes the connection even when the verification fails. The REQUIRETLS CHAIN and DANE options allow the message sender to specify that successful certificate validation, using either or both of two different methods, is required before sending the message.

Another active attack involves the spoofing of DNS MX records of the recipient domain. An attacker having this capability could cause the message to be redirected to a mail server under the attacker's own control, which would presumably have a valid certificate. The REQUIRETLS DNSSEC option allows the message sender to require that valid DNSSEC [RFC4033] signatures be obtained when locating the recipient's mail server, in order to address that attack.

In addition to support of the DNSSEC option, domains receiving email SHOULD deploy DNSSEC and SMTP clients SHOULD deploy DNSSEC verification.

7.3. Bad Actor MTAs

A bad-actor MTA along the message transmission path could misrepresent its support of REQUIRETLS and/or actively strip REQUIRETLS tags from messages it handles. However, since intermediate MTAs are already trusted with the cleartext of messages

they handle, and are not part of the threat model for transport-layer security, they are also not part of the threat model for REQUIRETLS.

It should be reemphasized that since SMTP TLS is a transport-layer security protocol, messages sent using REQUIRETLS are not encrypted end-to-end and are visible to MTAs that are part of the message delivery path. Messages containing sensitive information that MTAs should not have access to MUST be sent using end-to-end content encryption such as OpenPGP [RFC4880] or S/MIME [RFC5751].

8. Acknowledgements

The author would like to acknowledge many helpful suggestions on the ietf-smtp and uta mailing lists, in particular those of Viktor Dukhovni, Tony Finch, Jeremy Harris, Arvel Hathcock, John Klensin, John Levine, Rolf Sonneveld, and Per Thorsheim.

9. Revision History

To be removed by RFC Editor upon publication as an RFC.

9.1. Changes Since -02 Draft

- o Incorporation of "MAY TLS" functionality as REQUIRETLS=NO per suggestion on UTA WG mailing list.
- o Additional guidance on bounce messages

9.2. Changes Since -01 Draft

- o Specified retries when multiple MX hosts exist for a given domain.
- o Clarified generation of non-delivery messages
- o Specified requirements for application of REQUIRETLS to mail forwarders and mailing lists.
- o Clarified DNSSEC requirements to include MX lookup only.
- o Corrected terminology regarding message retrieval vs. delivery.
- o Changed category to standards track.

9.3. Changes Since -00 Draft

- o Conversion of REQUIRETLS from an SMTP verb to a MAIL FROM parameter to better associate REQUIRETLS requirements with transmission of individual messages.

- o Addition of an option to require DNSSEC lookup of the remote mail server, since this affects the common name of the certificate that is presented.
- o Clarified the wording to more clearly state that TLS sessions must be established and not simply that STARTTLS is negotiated.
- o Introduced need for minimum encryption standards (key lengths and algorithms)
- o Substantially rewritten Security Considerations section

10. References

10.1. Normative References

[MailParams]

Internet Assigned Numbers Authority (IANA), "IANA Mail Parameters", 2007,
<<http://www.iana.org/assignments/mail-parameters>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997,
<<http://www.rfc-editor.org/info/rfc2119>>.

[RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, DOI 10.17487/RFC3207, February 2002, <<http://www.rfc-editor.org/info/rfc3207>>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005,
<<http://www.rfc-editor.org/info/rfc4035>>.

[RFC5248] Hansen, T. and J. Klensin, "A Registry for SMTP Enhanced Mail System Status Codes", BCP 138, RFC 5248, DOI 10.17487/RFC5248, June 2008,
<<http://www.rfc-editor.org/info/rfc5248>>.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008,
<<http://www.rfc-editor.org/info/rfc5321>>.

[SMTPStatusCodes]

Internet Assigned Numbers Authority (IANA), "Simple Mail Transfer Protocol (SMTP) Enhanced Status Codes Registry", 2008, <<http://www.iana.org/assignments/smtp-enhanced-status-codes>>.

10.2. Informative References

[I-D.ietf-uta-mta-sts]

Margolis, D., Risher, M., Ramakrishnan, B., Brotman, A., and J. Jones, "SMTP MTA Strict Transport Security (MTA-STS)", draft-ietf-uta-mta-sts-02 (work in progress), December 2016.

[RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, DOI 10.17487/RFC1939, May 1996, <<http://www.rfc-editor.org/info/rfc1939>>.

[RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, DOI 10.17487/RFC3501, March 2003, <<http://www.rfc-editor.org/info/rfc3501>>.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<http://www.rfc-editor.org/info/rfc4033>>.

[RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, DOI 10.17487/RFC4880, November 2007, <<http://www.rfc-editor.org/info/rfc4880>>.

[RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, DOI 10.17487/RFC5751, January 2010, <<http://www.rfc-editor.org/info/rfc5751>>.

[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, DOI 10.17487/RFC6125, March 2011, <<http://www.rfc-editor.org/info/rfc6125>>.

[RFC7672] Dukhovni, V. and W. Hardaker, "SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS)", RFC 7672, DOI 10.17487/RFC7672, October 2015, <<http://www.rfc-editor.org/info/rfc7672>>.

Author's Address

Jim Fenton
Altmode Networks
704 Benvenue Avenue
Los Altos, California 94024
USA

Email: fenton@bluepopcorn.net

Network Working Group
Internet-Draft
Updates: 1939, 2595, 3464, 3501, 5068,
6186, 6409 (if approved)
Intended status: Standards Track
Expires: June 9, 2018

K. Moore
Windrock, Inc.
C. Newman
Oracle
December 6, 2017

Cleartext Considered Obsolete: Use of TLS for Email Submission and
Access
draft-ietf-uta-email-deep-12

Abstract

This specification outlines current recommendations for the use of Transport Layer Security (TLS) to provide confidentiality of email traffic between a mail user agent (MUA) and a mail submission or mail access server.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on June 9, 2018.

Copyright Notice

Copyright (c) 2017 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 (<https://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
2.	Conventions and Terminology Used in This Document	3
3.	Implicit TLS	4
3.1.	Implicit TLS for POP	5
3.2.	Implicit TLS for IMAP	5
3.3.	Implicit TLS for SMTP Submission	5
3.4.	Implicit TLS Connection Closure for POP, IMAP and SMTP Submission	6
4.	Use of TLS by Mail Access Services and Message Submission Services	6
4.1.	Deprecation of Services Using Cleartext and TLS Versions < 1.1	8
4.2.	Mail Server Use of Client Certificate Authentication	9
4.3.	Recording TLS Cipher Suite in Received Header	9
4.4.	TLS Server Certificate Requirements	10
4.5.	Recommended DNS records for mail protocol servers	10
4.5.1.	MX records	10
4.5.2.	SRV records	10
4.5.3.	DNSSEC	10
4.5.4.	TLSA records	11
4.6.	Changes to Internet Facing Servers	11
5.	Use of TLS by Mail User Agents	11
5.1.	Use of SRV records in Establishing Configuration	12
5.2.	Minimum Confidentiality Level	13
5.3.	Certificiate Validation	14
5.4.	Certificate Pinning	15
5.5.	Client Certificate Authentication	15
6.	Considerations related to Anti-Virus/Anti-Spam Software and Services	16
7.	IANA Considerations	16
7.1.	POP3S Port Registration Update	17
7.2.	IMAPS Port Registration Update	17
7.3.	Submissions Port Registration	17
7.4.	Additional registered clauses for Received fields	18
8.	Security Considerations	18
9.	References	19
9.1.	Normative References	19
9.2.	Informative References	21
Appendix A. Design Considerations		22
Appendix B. Change Log		24
Appendix C. Acknowledgements		29
Authors' Addresses		29

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) Submission [RFC6409] usually has Transport Layer Security (TLS) [RFC5246] support but often does not use it in a way that maximizes end-user confidentiality. This specification describes current recommendations for the use of TLS in interactions between Mail User Agents and Mail Access Services, and between Mail User Agents and Mail Submission Services.

In brief, this memo now recommends that:

- o TLS version 1.2 or greater be used for all traffic between mail user agents (MUAs) and mail submission servers, and also between MUAs and mail access servers.
- o MUAs and mail service providers discourage use of cleartext protocols for mail access and mail submission, and deprecate use of cleartext protocols for these purposes as soon as practicable.
- o Use of "Implicit TLS" on ports reserved for that purpose, in preference to STARTTLS on a port that otherwise supports cleartext.

This memo does not address use of TLS with SMTP for message relay (where Message Submission [RFC6409] does not apply). Improved use of TLS with SMTP for message relay requires a different approach. One approach to address that topic is described in [RFC7672]; another is in [I-D.ietf-uta-mta-sts].

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.

2. Conventions and Terminology Used in This Document

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 BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

The term "Implicit TLS" refers to the automatic negotiation of TLS whenever a TCP connection is made on a particular TCP port that is used exclusively by that server for TLS connections. The term "Implicit TLS" is intended to contrast with use of STARTTLS and similar commands in POP, IMAP, SMTP message submission, and other

protocols, that are used by client and server to explicitly negotiate TLS on an established cleartext TCP connection.

The term "Mail Access Services" includes POP, IMAP and any other protocol used to access or modify received messages, or to access or modify a mail user's account configuration.

"Mail Submission Service" refers to the use of the protocol specified in [RFC6409] (or one of its predecessors or successors) for submission of outgoing messages for delivery to recipients.

The term "Mail Service Provider" (MSP) refers to a provider of Mail Access Services and/or Mail Submission Services.

The term "Mail Account" refers to a user's identity with a Mail Service Provider, that user's authentication credentials, any user email that is stored by the MSP, and any other per-user configuration information maintained by the MSP (for example, spam filtering instructions). Most Mail User Agents (MUAs) support the ability to access multiple Mail Accounts.

For each account that an MUA accesses on its user's behalf, it must have the server names, ports, authentication credentials, and other configuration information specified by the user. This information which is used by the MUA is referred to as "Mail Account Configuration"

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].

3. 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 cleartext 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 encourage more widespread use of TLS, and to encourage a greater consistency for how TLS is used, this specification now recommends use of Implicit TLS for POP, IMAP, SMTP Submission, and all other protocols used between a Mail User Agent and a mail service.

3.1. 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 [RFC7817]. 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 a client certificate was supplied during the TLS handshake.

See Section 5.5 and Section 4.2 for additional information on client certificate authentication. See Section 7.1 for port registration information.

3.2. 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 [RFC7817]. Once the TLS session is established, IMAP [RFC3501] protocol messages are exchanged as TLS application data for the remainder of the TCP connection. If a client certificate was 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 5.5 and Section 4.2 for additional information on client certificate authentication. See Section 7.1 and Section 7.2 for port registration information.

3.3. 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 [RFC7817]. 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.)

The STARTTLS mechanism on port 587 is relatively widely deployed due to the situation with port 465 (discussed in Section 7.3). This differs from IMAP and POP services where Implicit TLS is more widely

deployed on servers than STARTTLS. It is desirable to migrate core protocols used by MUA software to Implicit TLS over time for consistency as well as the additional reasons discussed in Appendix A. However, to maximize use of encryption for submission it is desirable to support both mechanisms for Message Submission over TLS for a transition period of several years. As a result, clients and servers SHOULD implement both STARTTLS on port 587 and Implicit TLS on port 465 for this transition period. Note that there is no significant difference between the security properties of STARTTLS on port 587 and Implicit TLS on port 465 if the implementations are correct and both client and server are configured to require successful negotiation of TLS prior to message submission.

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 5.5 and Section 4.2 for additional information on client certificate authentication. See Section 7.3 for port registration information.

3.4. Implicit TLS Connection Closure for POP, IMAP and SMTP Submission

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 (e.g. call the close() function on the TCP socket or otherwise issue a TCP CLOSE ([RFC0793] section 3.5) without waiting for a TLS response from the server.

4. Use of TLS by Mail Access Services and Message Submission Services

The following requirements and recommendations apply to Mail Access Services and Mail Submission Services:

- o Mail Service Providers (MSPs) that support POP, IMAP, and/or Message Submission, MUST support TLS access for those services.
- o Other services than POP, IMAP and/or Message Submission provided by MSPs SHOULD support TLS access, and MUST support TLS access for those services which support authentication via username and password.
- o MSPs that support POP, IMAP, and/or Message Submission, SHOULD provide and support instances of those services which use Implicit TLS. (See Section 3.)

- o For compatibility with existing MUAs and existing MUA configurations, MSPs SHOULD also, in the near term, provide instances of these services which support STARTTLS. This will permit legacy MUAs to discover new availability of TLS capability on servers, and may increase use of TLS by such MUAs. However, servers SHOULD NOT advertise STARTTLS if use of the STARTTLS command by a client is likely to fail (for example, if the server has no server certificate configured.)
- o MSPs SHOULD advertise their Mail Access Services and Mail Submission Services using DNS SRV records according to [RFC6186]. (In addition to making correct configuration easier for MUAs, this provides a way by which MUAs can discover when an MSP begins to offer TLS-based services.) Services supporting TLS SHOULD be advertised in preference to cleartext services (if offered). In addition, services using Implicit TLS SHOULD be advertised in preference to services supporting STARTTLS (if offered). (See also Section 4.5.)
- o MSPs SHOULD deprecate use of cleartext Mail Access Services and Mail Submission Services as soon as practicable. (See Section 4.1.)
- o MSPs currently supporting such use of cleartext SMTP (on port 25) as a means of message submission by their users (whether or not requiring authentication) SHOULD transition their users to using TLS (either Implicit TLS or STARTTLS) as soon as practicable.
- o Mail services MUST support TLS 1.2 or later.
- o All Mail services SHOULD implement the recommended TLS cipher suites described in [RFC7525] or a future BCP or standards track revision of that document.
- o Mail services currently supporting SSL 2.x, SSL 3.0, or TLS 1.0 SHOULD transition their users to later versions of TLS, and discontinue support for those versions of SSL and TLS, as soon as practicable.
- o Mail Submission Servers accepting mail using TLS SHOULD include the TLS ciphersuite of the session in which the mail was received, in the Received field of the outgoing message. (See Section 4.3.)
- o All Mail services implementing TLS SHOULD log TLS cipher information along with any connection or authentication logs that they maintain.

Additional considerations and details appear below.

4.1. Deprecation of Services Using Cleartext and TLS Versions < 1.1

The specific means employed for deprecation of cleartext Mail Access Services and Mail Submission Services MAY vary from one MSP to the next in light of their user communities' needs and constraints. For example, an MSP MAY implement a gradual transition in which, over time, more and more users are forbidden to authenticate to cleartext instances of these services, thus encouraging those users to migrate to Implicit TLS. Access to cleartext services should eventually be either disabled, or limited strictly for use by legacy systems which cannot be upgraded.

After a user's ability to authenticate to a service using cleartext is revoked, the server denying such access MUST NOT provide any indication over a cleartext channel of whether the user's authentication credentials were valid. An attempt to authenticate as such a user using either invalid credentials or valid credentials MUST both result in the same indication of access being denied.

Also, users previously authenticating with passwords sent as cleartext SHOULD be required to change those passwords when migrating to TLS, if the old passwords were likely to have been compromised. (For any large community of users using public Internet to access mail without encryption, compromise of at least some of those passwords should be assumed.)

Transition of users from SSL or TLS 1.0 to later versions of TLS MAY be accomplished by a means similar to that described above. There are multiple ways to accomplish this. One way is for the server to refuse a ClientHello message from any client sending a ClientHello.version field corresponding to any version of SSL or TLS 1.0. Another way is for the server to accept ClientHello messages from some client versions that it does not wish to support, but later refuse to allow the user to authenticate. The latter method may provide a better indication to the user of the reason for the failure but (depending on the protocol and method of authentication used) may also risk exposure of the user's password over a channel which is known to not provide adequate confidentiality.

It is RECOMMENDED that new users be required to use TLS version 1.1 or greater from the start. However an MSP may find it necessary to make exceptions to accommodate some legacy systems which support only earlier versions of TLS, or only cleartext.

4.2. Mail Server Use of Client Certificate Authentication

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.

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.

4.3. 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 confidentiality or security. The TLS cipher suite provides additional information about the level of 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. This clause SHOULD be included whenever a Submission server generates a Received header field for a message received via TLS. The value included in this additional clause SHOULD be the registered cipher suite name (e.g., TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256) 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. In addition, the Diffie-Hellman group name associated with the ciphersuite MAY be included (when applicable and known) following the ciphersuite name. The ABNF for the field follows:

```
tls-cipher-clause = CFWS "tls" FWS tls-cipher [ CFWS "group" FWS dh-group ]
tls-cipher         = tls-cipher-name / tls-cipher-hex
tls-cipher-name    = ALPHA *(ALPHA / DIGIT / "_")
; as registered in IANA cipher suite registry
tls-cipher-hex     = "0x" 4HEXDIG
dh-group           = ALPHA *(ALPHA / DIGIT / "_" / "-")
; as registered in IANA TLS Supported Groups Registry
```

4.4. TLS Server Certificate Requirements

MSPs MUST maintain valid server certificates for all servers. See [RFC7817] for the recommendations and requirements necessary to achieve this.

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 certificate 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]. Mail servers supporting SNI need to support the post-SRV hostname to interoperate with MUAs that have not implemented RFC 6186. For more discussion of this problem, see section 5.1 of [RFC7817].

4.5. 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.

4.5.1. 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 [RFC4033]. This is mentioned here only for completeness, as handling of inbound mail is out of scope for this document.

4.5.2. 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.

4.5.3. 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 if and when the parent DNS zone supports doing so.

4.5.4. 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.

4.6. Changes to Internet Facing Servers

When an MSP changes the Internet Facing Servers providing mail access and mail submission services, including SMTP-based spam/virus filters, it is generally necessary to support the same and/or a newer version of TLS and the same security directives that were previously advertised.

5. Use of TLS by Mail User Agents

The following requirements and recommendations apply to Mail User Agents:

- o MUAs SHOULD be capable of using DNS SRV records to discover Mail Access Services and Mail Submission Services that are advertised by a MSP for an account being configured. Other means of discovering server configuration information (e.g. a database maintained by the MUA vendor) MAY also be supported. (See Section 5.1 for more information.)
- o MUAs SHOULD be configurable to require a minimum level of confidentiality for any particular Mail Account, and refuse to exchange information via any service associated with that Mail Account if the session does not provide that minimum level of confidentiality. (See Section 5.2.)
- o MUAs MUST NOT treat a session as meeting a minimum level of confidentiality if the server's TLS certificate cannot be validated. (See Section 5.3.)
- o MUAs MAY impose other minimum confidentiality requirements in the future, e.g. in order to discourage use of TLS versions or cryptographic algorithms in which weaknesses have been discovered.
- o MUAs SHOULD provide a prominent indication of the level of confidentiality associated with an account configuration that is appropriate for the user interface (for example, a "lock" icon or changed background color for a visual interface, or some sort of audible indication for an audio user interface), at appropriate times and/or locations in order to inform the user of the confidentiality of the communications associated with that

account. For example, this might be done whenever (a) prompting the user for authentication credentials, (b) the user is composing mail that will be sent to a particular submission server, (c) a list of accounts is displayed (particularly if the user can select from that list to read mail), or (d) the user is requesting to view or update any configuration data that will be stored on a remote server. If, however, an MUA provides such an indication, it **MUST NOT** indicate confidentiality for any connection that does not at least use TLS 1.1 with certificate verification and also meet the minimum confidentiality requirements associated with that account.

- o MUAs **MUST** implement TLS 1.2 [RFC5246] or later. Earlier TLS and SSL versions **MAY** also be supported so long as the MUA requires at least TLS 1.1 [RFC4346] when accessing accounts that are configured to impose minimum confidentiality requirements.
- o All MUAs **SHOULD** implement the recommended TLS cipher suites described in [RFC7525] or a future BCP or standards track revision of that document.
- o MUAs that are configured to not require minimum confidentiality for one or more accounts **SHOULD** detect when TLS becomes available on those accounts (using [RFC6186] or other means), and offer to upgrade the account to require TLS.

Additional considerations and details appear below.

5.1. 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** ignore advertised services that do not satisfy minimum confidentiality requirements, unless the user has explicitly requested reduced confidentiality. This will have the effect of causing the MUA to default to ignoring advertised configurations that 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.

Similarly, a MUA MUST NOT attempt to "test" a particular mail account configuration by submitting the user's authentication credentials to a server, unless a TLS session meeting minimum confidentiality levels has been established with that server. If minimum confidentiality requirements have not been satisfied, the MUA must explicitly warn the user that his password may be exposed to attackers before testing the new configuration.

When establishing a new configuration for connecting to an IMAP, POP, or SMTP submission server, based on SRV records, an MUA SHOULD either verify that the SRV records are signed using DNSSEC, or that the target FQDN of the SRV record matches the original server FQDN for which the SRV queries were made. If the target FQDN is not in the queried domain, the MUA SHOULD verify with the user that the SRV target FQDN is suitable for use, before executing any connections to the host. (See [RFC6186] section 6).

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.

5.2. Minimum Confidentiality Level

MUAs SHOULD, by default, require a minimum level of confidentiality for services accessed by each account. For MUAs supporting the ability to access multiple mail accounts, this requirement SHOULD be configurable on a per-account basis.

The default minimum expected level of confidentiality for all new accounts MUST require successful validation of the server's certificate and SHOULD require negotiation of TLS version 1.1 or greater. (Future revisions to this specification may raise these requirements or impose additional requirements to address newly-discovered weaknesses in protocols or cryptographic algorithms.)

MUAs MAY permit the user to disable this minimum confidentiality requirement during initial account configuration, or subsequently editing an account configuration, but MUST warn users that such a configuration will not assure privacy for either passwords or messages.

An MUA which is configured to require a minimum level of confidentiality for a mail account MUST NOT attempt to perform any operation other than capability discovery, or STARTTLS for servers not using Implicit TLS, unless the minimum level of confidentiality is provided by that connection.

MUAs SHOULD NOT allow users to easily access or send mail via an connection, or authenticate to any service using a password, if that account is configured to impose minimum confidentiality requirements and that connection does not meet all of those requirements. An example of "easily access" would be to display a dialog informing the user that the security requirements of the account were not met by the connection, but allowing the user to "click through" to send mail or access the service anyway. Experience indicates that users presented with such an option often "click through" without understanding the risks that they're accepting by doing so. Furthermore, users who frequently find the need to "click through" to use an insecure connection may become conditioned to do so as a matter of habit, before considering whether the risks are reasonable in each specific instance.

An MUA which is not configured to require a minimum level of confidentiality for a mail account SHOULD still attempt to connect to the services associated with that account using the most secure means available, e.g. by using Implicit TLS or STARTTLS.

5.3. Certificate Validation

MUAs MUST validate TLS server certificates according to [RFC7817] and PKIX [RFC5280].

MUAs MAY also support DANE [RFC6698] as a means of validating server certificates in order to meet minimum confidentiality requirements.

MUAs MAY support use of certificate pinning but MUST NOT consider a connection in which the server's authenticity relies on certificate pinning, as providing the minimum level of confidentiality. (See Section 5.4.)

5.4. Certificate Pinning

During account setup, the MUA will identify servers that provide account services such as mail access and mail submission (the previous section describes one way to do this). The certificates for these servers are verified using the rules described in [RFC7817] and PKIX [RFC5280]. In the event the certificate does not validate due to an expired certificate, lack of appropriate chain of trust, or lack of identifier match, the MUA MAY offer to create a persistent binding between that certificate and the saved host name for the server, for use when accessing that account's servers. This is called certificate pinning.

(Note: This use of the term "certificate pinning" means something subtly different than "HTTP Public Key Pinning" [RFC7469]. The dual use of the same term is confusing, but unfortunately both uses are well-established.)

Certificate pinning is only appropriate during mail account setup and MUST NOT be offered as an option in response to a failed certificate validation for an existing mail account. An MUA that allows certificate pinning MUST NOT allow a certificate pinned for one account to validate connections for other accounts. An MUA that allows certificate pinning MUST also allow a user to undo the pinning, i.e. to revoke trust in a certificate that has previously been pinned.

A pinned certificate is subject to a man-in-the-middle attack at account setup time, and typically lacks a mechanism to automatically revoke or securely refresh the certificate. Note also that a man-in-the-middle attack at account setup time will expose the user's password to the attacker (if a password is used). Therefore use of a pinned certificate does not meet the requirement for a minimum confidentiality level, and an MUA MUST NOT indicate to the user that the such confidentiality is provided. Additional advice on certificate pinning is present in [RFC6125].

5.5. Client Certificate Authentication

MUAs MAY implement client certificate authentication on the Implicit TLS port. An MUA MUST NOT provide a client certificate during the TLS handshake unless the server requests one and the MUA has been authorized to use that client certificate with that account. Having the end-user explicitly configure a client certificate for use with a given account is sufficient to meet this requirement. However, installing a client certificate for use with one account MUST NOT automatically authorize use of that certificate with other accounts. This is not intended to prohibit site-specific authorization

mechanisms, such as a site-administrator-controlled mechanism to authorize use of a client certificate with a given account, or a domain-name matching mechanism.

Note: The requirement that the server request a certificate is just a restatement of the TLS protocol rules, e.g. [RFC5246] section 7.4.6. The requirement that the client not send a certificate not known to be acceptable to the server is pragmatic in multiple ways: the current TLS protocol provides no way for the client to know which of potentially multiple certificates it should use; also, when the client sends a certificate it is potentially disclosing its identity (or its user's identity) to both the server and to any party with access to the transmission medium, perhaps unnecessarily and for no useful purpose.

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]).

6. Considerations related to 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, are out of scope for this specification. However, some services use an SMTP relay proxy that intercepts mail at the application layer to perform a scan and proxy or forward to another MTA. Deploying AVAS services in this way can cause many problems [RFC2979] including direct interference with this specification, and other forms of confidentiality or security reduction. An AVAS product or service is considered compatible with this specification if all IMAP, POP and SMTP-related software (including proxies) it includes are compliant with this specification.

Note that end-to-end email encryption prevents AVAS software and services from using email content as part of a spam or virus assessment. Furthermore, while a minimum confidentiality level can prevent a man-in-the-middle from introducing spam or virus content between the MUA and Submission server, it does not prevent other forms of client or account compromise. Use of AVAS services for submitted email therefore remains necessary.

7. IANA Considerations

7.1. 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)
Port Number: 995

7.2. IMAPS Port Registration Update

IANA is asked to update the registration of the TCP well-known port 993 using the following templates ([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)
Port Number: 993

No changes to existing UDP port assignments for pop3s or imaps are being requested.

7.3. Submissions Port Registration

IANA is asked to assign an alternate usage of TCP 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)
Port Number: 465

This is a one-time procedural exception to the rules in RFC 6335. This requires explicit IESG approval and does not set a precedent. Note: Since the purpose of this alternate usage assignment is to align with widespread existing practice, and there is no known usage

of UDP port 465 for message submission over TLS, IANA is not being asked to assign an alternate usage of UDP port 465.

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.

7.4. Additional registered clauses for Received fields

Per the provisions in [RFC5321], IANA is requested to add two additional-registered-clauses for Received fields as defined in Section 4.3 of this document:

- o "tls" indicating the TLS cipher used (if applicable), and
- o "group" indicating the Diffie-Hellman group used with the TLS cipher (if applicable)

The descriptions and syntax of these additional clauses are in Section 4.3 of this document.

8. Security Considerations

This entire document is about security considerations. In general, this is targeted to improve mail confidentiality 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.

Implementers should be aware that use of client certificates with TLS 1.2 reveals the user's identity to any party with ability to read packets from the transmission medium, and therefore may compromise

the user's privacy. There seems to be no easy fix with TLS 1.2 or earlier versions other than to avoid presenting client certificates except when there is explicit authorization to do so. TLS 1.3 [I-D.ietf-tls-tls13] appears to reduce the privacy risk somewhat.

9. References

9.1. Normative References

- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.
- [RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, DOI 10.17487/RFC1939, May 1996, <<https://www.rfc-editor.org/info/rfc1939>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, DOI 10.17487/RFC3207, February 2002, <<https://www.rfc-editor.org/info/rfc3207>>.
- [RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, DOI 10.17487/RFC3501, March 2003, <<https://www.rfc-editor.org/info/rfc3501>>.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<https://www.rfc-editor.org/info/rfc4033>>.
- [RFC5034] Siemborski, R. and A. Menon-Sen, "The Post Office Protocol (POP3) Simple Authentication and Security Layer (SASL) Authentication Mechanism", RFC 5034, DOI 10.17487/RFC5034, July 2007, <<https://www.rfc-editor.org/info/rfc5034>>.
- [RFC5068] Hutzler, C., Crocker, D., Resnick, P., Allman, E., and T. Finch, "Email Submission Operations: Access and Accountability Requirements", BCP 134, RFC 5068, DOI 10.17487/RFC5068, November 2007, <<https://www.rfc-editor.org/info/rfc5068>>.

- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [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, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <<https://www.rfc-editor.org/info/rfc5322>>.
- [RFC6186] Daboo, C., "Use of SRV Records for Locating Email Submission/Access Services", RFC 6186, DOI 10.17487/RFC6186, March 2011, <<https://www.rfc-editor.org/info/rfc6186>>.
- [RFC6409] Gellens, R. and J. Klensin, "Message Submission for Mail", STD 72, RFC 6409, DOI 10.17487/RFC6409, November 2011, <<https://www.rfc-editor.org/info/rfc6409>>.
- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <<https://www.rfc-editor.org/info/rfc6698>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.
- [RFC7672] Dukhovni, V. and W. Hardaker, "SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS)", RFC 7672, DOI 10.17487/RFC7672, October 2015, <<https://www.rfc-editor.org/info/rfc7672>>.

- [RFC7817] Melnikov, A., "Updated Transport Layer Security (TLS) Server Identity Check Procedure for Email-Related Protocols", RFC 7817, DOI 10.17487/RFC7817, March 2016, <<https://www.rfc-editor.org/info/rfc7817>>.

9.2. Informative References

- [I-D.ietf-tls-tls13] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", draft-ietf-tls-tls13-21 (work in progress), July 2017.
- [I-D.ietf-uta-mta-sts] Margolis, D., Risher, M., Ramakrishnan, B., Brotman, A., and J. Jones, "SMTP MTA Strict Transport Security (MTA-STS)", draft-ietf-uta-mta-sts-09 (work in progress), September 2017.
- [RFC2595] Newman, C., "Using TLS with IMAP, POP3 and ACAP", RFC 2595, DOI 10.17487/RFC2595, June 1999, <<https://www.rfc-editor.org/info/rfc2595>>.
- [RFC2979] Freed, N., "Behavior of and Requirements for Internet Firewalls", RFC 2979, DOI 10.17487/RFC2979, October 2000, <<https://www.rfc-editor.org/info/rfc2979>>.
- [RFC3848] Newman, C., "ESMTP and LMTP Transmission Types Registration", RFC 3848, DOI 10.17487/RFC3848, July 2004, <<https://www.rfc-editor.org/info/rfc3848>>.
- [RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, DOI 10.17487/RFC4346, April 2006, <<https://www.rfc-editor.org/info/rfc4346>>.
- [RFC4422] Melnikov, A., Ed. and K. Zeilenga, Ed., "Simple Authentication and Security Layer (SASL)", RFC 4422, DOI 10.17487/RFC4422, June 2006, <<https://www.rfc-editor.org/info/rfc4422>>.
- [RFC4954] Siemborski, R., Ed. and A. Melnikov, Ed., "SMTP Service Extension for Authentication", RFC 4954, DOI 10.17487/RFC4954, July 2007, <<https://www.rfc-editor.org/info/rfc4954>>.
- [RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <<https://www.rfc-editor.org/info/rfc5321>>.

- [RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <<https://www.rfc-editor.org/info/rfc6066>>.
- [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, DOI 10.17487/RFC6125, March 2011, <<https://www.rfc-editor.org/info/rfc6125>>.
- [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, DOI 10.17487/RFC6335, August 2011, <<https://www.rfc-editor.org/info/rfc6335>>.
- [RFC7469] Evans, C., Palmer, C., and R. Sleevi, "Public Key Pinning Extension for HTTP", RFC 7469, DOI 10.17487/RFC7469, April 2015, <<https://www.rfc-editor.org/info/rfc7469>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

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. (Also it's less correct than it used to be because "export" ciphersuites are no longer supported in modern versions of TLS.) 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 confidentiality facilities actually deployed is so important it should trump design purity considerations.

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. The remaining option is to document the current state of the world and support future use of port 465 for submission as this increases consistency and ease-of-deployment for TLS email submission.

Appendix B. Change Log

Changes since draft-ietf-uta-email-deep-07:

- o After discussion with the WG in Prague, removed BCP language and once again made unambiguous that this is intended as a standards-track document.
- o Server implementations now MUST implement TLS 1.2, consistent with RFC 7525. MUAs may still consider a TLS 1.1 session as meeting minimum confidentiality requirements.
- o MSPs now MUST support TLS for POP, IMAP, Submission, and any other services that use username/password authentication.
- o Added text to clarify the purpose of recommending that MSPs use DNS SRV records to advertise services.
- o Changed text about MUAs not blindly trusting unsigned SRV records, to instead restate RFC 6186 requirements.

Changes since draft-ietf-uta-email-deep-06:

- o On the recommendation of one of the co-chairs and some working group members, rewrote document with the intended status of BCP. This involved removing a great deal of text that consisted essentially of new protocol specification, especially the STS features, on the theory that a BCP should base its recommendations on current practice, and that new protocol features should be subject to the interoperability test requirements associated with normal standards-track documents.

Changes since draft-ietf-uta-email-deep-05:

- o Clarify throughout that the confidentiality assurance level associated with a mail account is a minimum level; attempt to distinguish this from the current confidentiality level provided by a connection between client and server.
- o Change naming for confidentiality assurance levels: instead of "high" or "no" confidence, assign numbers 1 and 0 to them respectively. This because it seems likely that in the not-too-distant future, what was defined in -05 as "high" confidence will be considered insufficient, and calling that "high" confidence will become misleading. For example, relying entirely on a list of trusted CAs to validate server certificates from arbitrary parties, appears to be less and less reliable in practice at thwarting MITM attacks.

- o Clarify that if some services associated with a mail account don't meet the minimum confidentiality assurance level assigned to that account, other services that do meet that minimum confidentiality assurance level may continue to be used.
- o Clarify that successful negotiation of at least TLS version 1.1 is required as a condition of meeting confidentiality assurance level 1.
- o Clarify that validation of a server certificate using either DANE or PKIX is sufficient to meet the certificate validation requirement of confidentiality assurance level 1.
- o Clarify that minimum confidentiality assurance levels are separate from security directives, and that the requirements of both mechanisms must be met.
- o Explicitly cite an example that a security directive of `tls-version=1.2` won't be saved if the currently negotiated `tls-version` is 1.1. (This example already appeared a bit later in the text, but for author KM it seemed to make the mechanism clearer to use this example earlier.)
- o Clarify some protocol examples as to whether PKIX or DANE was used to verify a server's certificate.
- o Remove most references to DEEP as the conversion from DEEP to MUA-STS seemed incomplete, but kept the DEEP command for use in POP3 on the assumption that author CN wanted it that way.
- o Removed most references to "latch" and derivative words.
- o Added `pkix+dane` as a value for the `tls-cert` directive, to indicate (from a server) that both PKIX and DANE validation will be supported, or (from a client) that both PKIX and DANE were used to validate a certificate. Also clarified what each of any, `pkix`, `dane`, and `pkix+dane` mean when advertised by a server and in particular that `tls-cert=any` provides no assurance of future PKIX verifiability in contrast to `tls-cert=pkix` or `tls-cert=pkix+dane`. It seemed important to support the ability to evolve to using multiple trust anchors for certificate validation, but also to allow servers to have the option to migrate from PKIX to DANE if that made sense for them. This change seemed less disruptive than either defining additional directives, or allowing multiple instances of the same directive with different values to appear in the same advertisement.

- o Clarify interaction of this specification with anti-virus / anti-spam mechanisms.

Changes since draft-ietf-uta-email-deep-04:

- o Swap sections 5.1 and 5.3 ("Email Security Tags" and "Server DEEP Status") as that order may aid understanding of the model. Also rewrote parts of these two sections to try to make the model clearer.
- o Add text about versioning of security tags to make the model clearer.
- o Add example of security tag upgrade.
- o Convert remaining mention of TLS 1.0 to TLS 1.1.
- o Change document title from DEEP to MUA STS to align with SMTP relay STS.
 - * Slight updates to abstract and introductions.
 - * Rename security latches/tags to security directives.
 - * Rename server DEEP status to STS policy.
 - * Change syntax to use directive-style HSTS syntax.
- o Make HSTS reference normative.
- o Remove SMTP DSN header as that belongs in SMTP relay STS document.

Changes since draft-ietf-uta-email-deep-03:

- o Add more references to ietf-uta-email-tls-certs in implementation requirements section.
- o Replace primary reference to RFC 6125 with ietf-uta-email-tls-certs, so move RFC 6125 to informative list for this specification.

Changes since draft-ietf-uta-email-deep-02:

- o Make reference to design considerations explicit rather than "elsewhere in this document".
- o Change provider requirement so SMTP submission services are separate from SMTP MTA services as opposed to the previous

phrasing that required the servers be separate (which is too restrictive).

- o Update DANE SMTP reference

Changes since draft-ietf-uta-email-deep-01:

- o Change text in tls11 and tls12 registrations to clarify certificate rules, including additional PKIX and DANE references.
- o Change from tls10 to tls11 (including reference) as the minimum.
- o Fix typo in example 5.
- o Remove open issues section; enough time has passed so not worth waiting for more input.

Changes since draft-ietf-uta-email-deep-00:

- o Update and clarify abstract
- o use term confidentiality instead of privacy in most cases.
- o update open issues to request input for missing text.
- o move certificate pinning sub-section to account setup section and attempt to define it more precisely.
- o Add note about end-to-end encryption in AVAS section.
- o swap order of DNSSEC and TLSA sub-sections.
- o change meaning of 'tls10' and 'tls12' latches to require certificate validation.
- o Replace cipher suite advice with reference to RFC 7525. Change examples to use TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as cipher suite.
- o Add text to update IMAP, POP3 and Message Submission standards with newer TLS advice.
- o Add clearer text in introduction that this does not cover SMTP relay.
- o Update references to uta-tls-certs.

- o Add paragraph to Implicit TLS for SMTP Submission section recommending that STARTTLS also be implemented.

Changes since draft-newman-email-deep-02:

- o Changed "privacy assurance" to "confidentiality assurance"
- o Changed "low privacy assurance" to "no confidentiality assurance"
- o Attempt to improve definition of confidentiality assurance level.
- o Add SHOULD indicate when MUA is showing list of mail accounts.
- o Add SHOULD NOT latch tls10, tls12 tags until TLS negotiated.
- o Removed sentence about deleting and re-creating the account in latch failure section.
- o Remove use of word "fallback" with respect to TLS version negotiation.
- o Added bullet about changes to Internet facing servers to MSP section.
- o minor wording improvements based on feedback

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 C. Acknowledgements

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 Russ Housley, Alexey Melnikov and Dan Newman for review feedback. Thanks to Paul Hoffman for interesting feedback in initial conversations about this idea.

Authors' Addresses

Keith Moore
Windrock, Inc.
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

Using TLS in Applications
Internet-Draft
Intended status: Standards Track
Expires: September 19, 2016

D. Margolis
M. Risher
N. Lidzborski
W. Chuang
B. Long
Google, Inc
B. Ramakrishnan
Yahoo!, Inc
A. Brotman
Comcast, Inc
J. Jones
Microsoft, Inc
F. Martin
LinkedIn
K. Umbach
M. Laber
1&1 Mail & Media Development & Technology GmbH
March 18, 2016

SMTP Strict Transport Security
draft-margolis-smtp-sts-00

Abstract

SMTP STS is a mechanism enabling mail service providers to declare their ability to receive TLS-secured connections, to declare particular methods for certificate validation, and to request sending SMTP servers to report upon and/or refuse to deliver messages that cannot be delivered securely.

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 19, 2016.

Copyright Notice

Copyright (c) 2016 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. Terminology	3
2. Related Technologies	4
2.1. Differences from DANE	4
2.2. Advantages When Used with DANE	4
2.3. Advantages When Used Without DANE	4
2.4. Disadvantages When Used Without DANE	5
3. Policy Semantics	5
3.1. Formal Definition	6
3.2. Policy Expirations	8
3.3. Policy Authentication	8
3.4. Policy Validation	9
3.5. Policy Application	9
4. Failure Reporting	10
5. IANA Considerations	12
6. Security Considerations	12
7. Future Work	13
8. Appendix 1: Validation Pseudocode	14
9. Appendix 2: Domain Owner STS example record	14
10. Appendix 3: XML Schema for Failure Reports	14
11. Appendix 4: Example report	16
12. Normative References	17
Authors' Addresses	18

1. Introduction

The STARTTLS extension to SMTP [RFC3207] allows SMTP clients and hosts to establish secure SMTP sessions over TLS. In its current form, however, it fails to provide (a) message confidentiality -- because opportunistic STARTTLS is subject to downgrade attacks -- and

(b) server authenticity -- because the trust relationship from email domain to MTA server identity is not cryptographically validated.

While such "opportunistic" encryption protocols provide a high barrier against passive man-in-the-middle traffic interception, any attacker who can delete parts of the SMTP session (such as the "250 STARTTLS" response) or who can redirect the entire SMTP session (perhaps by overwriting the resolved MX record of the delivery domain) can perform such a downgrade or interception attack.

This document defines a mechanism for recipient domains to publish policies specifying:

- o whether MTAs sending mail to this domain can expect TLS support
- o how MTAs can validate the TLS server certificate presented during mail delivery
- o what an implementing sender should do with messages when TLS cannot be successfully negotiated

The mechanism described is separated into four logical components:

1. policy semantics: whether senders can expect a server for the recipient domain to support TLS encryption and how to validate the TLS certificate presented
2. policy authentication: how to determine the authenticity of a published policy delivered via DNS
3. failure report format: a mechanism for informing recipient domains about aggregate failure statistics
4. failure handling: what sending MTAs should do in the case of policy failures

1.1. Terminology

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

We also define the following terms for further use in this document:

- o STS Policy: A definition of the expected TLS availability and behavior, as well as the desired actions for a given domain when a sending MTA encounters different results.

- o Policy Domain: The domain against which an STS Policy is defined.

2. Related Technologies

The DANE TLSA record [RFC7672] is similar, in that DANE is also designed to upgrade opportunistic encryption into required encryption. DANE requires DNSSEC [RFC4033] for the secure delivery of policies; the mechanism described here presents a variant for systems not yet supporting DNSSEC, and specifies a method for reporting TLS negotiation failures.

2.1. Differences from DANE

The primary difference between the mechanism described here and DANE is that DANE requires the use of DNSSEC to authenticate DANE TLSA records, whereas SMTP STS relies on the certificate authority (CA) system and a trust-on-first-use (TOFU) approach to avoid interception. The TOFU model allows a degree of security similar to that of HPKP [RFC7469], reducing the complexity but without the guarantees on first use offered by DNSSEC. (For a thorough discussion of this trade-off, see the section `_Security_` `_Considerations_`.)

In addition, SMTP STS introduces a mechanism for failure reporting and a report-only mode, enabling progressive roll-out and auditing for compliance.

2.2. Advantages When Used with DANE

SMTP STS can be deployed for a recipient domain that also publishes a DANE TLSA record for SMTP. In these cases, the SMTP STS policy can additionally declare a process for failure reporting.

2.3. Advantages When Used Without DANE

When deployed without a DANE TLSA record, SMTP STS offers the following advantages compared to DANE:

- o `_Infrastructure_`: In comparison to DANE, this proposal does not require DNSSEC be deployed on either the sending or receiving domain. In addition, the reporting feature of SMTP STS can be deployed to perform offline analysis of STARTTLS failures, enabling mail providers to gain insight into the security of their SMTP connections without the need to modify MTA codebases directly.
- o `_Incrementalism_`: DANE does not provide a reporting mechanism and does not have a concept of "report-only" for failures; as a

result, a service provider has no choice but to "flip the switch" and affect the entire mail stream at once.

2.4. Disadvantages When Used Without DANE

When deployed alone (i.e. without a DANE record, and using Web PKI for certificate verification), SMTP STS offers the following disadvantages compared to DANE:

- o Infrastructure: DANE may be easier for some providers to deploy. In particular, for providers who already support DNSSEC, SMTP STS would additionally require they obtain a CA-signed x509 certificate for the recipient domain.
- o Security: DANE offers an advantage against policy-lookup DoS attacks; that is, while a DNSSEC-signed NX response to a DANE lookup authoritatively indicates the lack of a DANE record, such an option to authenticate policy non-existence does not exist when looking up a policy over plain DNS.

3. Policy Semantics

SMTP STS policies are distributed at the Policy Domain either through a new resource record, or as TXT records (similar to DMARC policies) under the name "_smtp_sts." (Current implementations deploy as TXT records.) For example, for the Policy Domain "example.com", the recipient's SMTP STS policy can be retrieved from "_smtp_sts.example.com."

(Future implementations may move to alternate methods of policy discovery or distribution. See the section `_Future_ _Work_` for more discussion.)

Policies MUST specify the following fields:

- o v: Version (plain-text, required). Currently only "STS1" is supported.
- o to: TLS-Only (plain-text, required). If "true," the receiving MTA requests that messages be delivered only if they conform to the STS policy. If "false," the receiving MTA requests that failure reports be delivered, as specified by the "rua" parameter.
- o mx: MX patterns (comma-separated list of plain-text MX match patterns, required). One or more comma-separated patterns matching the expected MX for this domain. For example, "_example.com,_.example.net" indicates that mail for this domain

might be handled by any MX whose hostname is a subdomain of "example.com" or "example.net."

- o a: The mechanism to use to authenticate this policy itself. See the section `_Policy_ _Authentication_` for more details. Possible values are:
 - * `webpki:URI`, where URI points to an HTTPS resource at the recipient domain that serves the same policy text.
 - * `dnssec`: Indicating that the policy is expected to be served over DNSSEC.
- o c: Constraints on the recipient MX's TLS certificate (plain-text, required). See the section `_Policy_ _Validation_` for more details. Possible values are:
 - * `webpki`: Indicating that the TLS certificate presented by the recipient MX will be validated according to the "web PKI" mechanism.
 - * `tlsa`: Indicating that the TLS certificate presented by the recipient MX will match a (presumed to exist) DANE TLSA record.
- o e: Max lifetime of the policy (plain-text integer seconds). Well-behaved clients SHOULD cache a policy for up to this value from last policy fetch time.
- o rua: Address to which aggregate feedback MAY be sent (comma-separated plain-text list of email addresses, optional). For example, "mailto:postmaster@example.com" from [RFC3986].

3.1. Formal Definition

The formal definition of the SMTP STS format, using [RFC5234], is as follows:

```

sts-uri      = URI [ "!" 1*DIGIT [ "k" / "m" / "g" / "t" ] ]
               ; "URI" is imported from [RFC3986]; commas (ASCII
               ; 0x2C) and exclamation points (ASCII 0x21)
               ; MUST be encoded; the numeric portion MUST fit
               ; within an unsigned 64-bit integer

sts-record   = sts-version sts-sep sts-to
               [sts-sep sts-mx]
               [sts-sep sts-a]
               [sts-sep sts-c]
               [sts-sep sts-e]
               [sts-sep sts-auri]
               [sts-sep]
               ; components other than sts-version and
               ; sts-to may appear in any order

sts-version  = "v" *WSP "=" *WSP %x53 %x54 %x53 %x31

sts-sep      = *WSP %x3b *WSP

sts-to       = "to" *WSP "=" *WSP ( "true" / "false" )

sts-mx       = "mx" *WSP "=" *WSP sts-domain-list

sts-domain-list = (domain-match *(", " domain-match))

domain-match  = [ "*" "." ] 1*dtext *(", " 1*dtext)

dtext         = %d30-39 /           ; 0-9
               %d41-5A /           ; a-z
               %61-7A /           ; A-Z
               %2D                 ; "-"

sts-a         = "a" *WSP "=" *WSP ( URI / "dnssec" )

sts-c         = "c" *WSP "=" *WSP ( "webpki" / "tlsa" )

sts-e         = "e" *WSP "=" *WSP 1*6DIGIT

sts-auri      = "rua" *WSP "=" *WSP
               sts-uri *( *WSP ", " *WSP sts-uri )

```

A size limitation in a sts-uri, if provided, is interpreted as a count of units followed by an OPTIONAL unit size ("k" for kilobytes, "m" for megabytes, "g" for gigabytes, "t" for terabytes). Without a unit, the number is presumed to be a basic byte count. Note that the units are considered to be powers of two; a kilobyte is 2¹⁰, a megabyte is 2²⁰, etc.

3.2. Policy Expirations

In order to resist attackers inserting a fraudulent policy, SMTP STS policies are designed to be long-lived, with an expiry typically greater than two weeks. Policy validity is controlled by two separate expiration times: the lifetime indicated in the policy ("e=") and the TTL on the DNS record itself. The policy expiration will ordinarily be longer than that of the DNS TTL, and senders SHOULD cache a policy (and apply it to all mail to the recipient domain) until the policy expiration.

An important consideration for domains publishing a policy is that senders will see a policy expiration as relative to the fetch of a policy cached by their recursive resolver. Consequently, a sender MAY treat a policy as valid for up to {expiration time} + {DNS TTL}. Publishers SHOULD thus continue to expect senders to apply old policies for up to this duration.

3.3. Policy Authentication

The security of a domain implementing an SMTP STS policy against an active man-in-the-middle depends primarily upon the long-lived caching of policies. However, to allow recipient domains to safely serve new policies prior to the expiration of a cached policy, and to prevent long-term (either malicious or active) denials of service, it is important that senders are able to validate a new policy retrieved for a recipient domain. There are two supported mechanisms for policy validation:

- o Web PKI: In this mechanism, indicated by the "webpki" value of the "a" field, the sender fetches a HTTPS resource from the URI indicated. For example, a=webpki:<https://example.com/.well-known/smime/smime-current> indicates that the sender should fetch the resource <https://example.com/.well-known/smime/smime-current>. In order for the policy to be valid, the HTTP response body served at this resource MUST exactly match the policy initially loaded via the DNS TXT method, and MUST be served from an HTTPS endpoint at the domain matching that of the recipient domain. (As this RFC progresses, the authors intend to register .well-known/smime-sts. See [RFC5785]. See Future Work for more information.)
- o DNSSEC: In this mechanism, indicated by the "dnssec" value of the "a" field, the sender MUST retrieve the policy via a DNSSEC signed response for the _smtp_sts TXT record.

When fetching a new policy when one is not already known, or when fetching a policy for a domain with an expired policy, unauthenticated policies MUST be trusted and honored. When fetching

a policy and authenticating it, as described in detail in `_Policy_Application_`, policies will be authenticated using the mechanism specified by the existing cached policy.

Note, however, as described in detail in `_Policy_Application_`, that new policies MUST NOT be considered as valid if they do not validate on first application. That is, a freshly fetched (and unused) policy that has not successfully been applied MUST be disregarded.

3.4. Policy Validation

When sending to an MX at a domain for which the sender has a valid and non-expired SMTP STS policy, a sending MTA honoring SMTP STS SHOULD validate that the recipient MX supports STARTTLS and offers a TLS certificate which is valid according to the semantics of the SMTP STS policy. Policies can specify certificate validity in one of two ways by setting the value of the "c" field in the policy description.

- o Web PKI: When the "c" field is set to "webpki", the certificate presented by the receiving MX MUST be valid for the MX name and chain to a root CA that is trusted by the sending MTA. The certificate MUST have a CN or SAN matching the MX hostname (as described in [RFC6125]) and be non-expired.
- o DANE TLSA: When the "c" field is set to "tlsa", the receiving MX MUST be covered by a DANE TLSA record for the recipient domain, and the presented certificate MUST be valid according to that record (as described by [RFC7672]).

A sending MTA who does not support the validation method required--for example, an MTA that does not have a DNSSEC-compatible resolver--MUST behave as though the policy did not validate. As described in the section on `_Policy_Application_`, a policy which has not ever been successfully validated MUST not be used to reject mail.

3.5. Policy Application

When sending to an MX at a domain for which the sender has a valid non-expired SMTP STS policy, a sending MTA honoring SMTP STS MAY apply the result of a policy validation one of two ways:

- o Report-only: In this mode, sending MTAs merely send a report to the designated report address indicating policy application failures. This can be done "offline", i.e. based on the MTA logs, and is thus a suitable low-risk option for MTAs who wish to enhance transparency of TLS tampering without making complicated changes to production mail-handling infrastructure.

- o Enforced: In this mode, sending MTAs SHOULD treat STS policy failures, in which the policy action is "reject", as a mail delivery error, and SHOULD terminate the SMTP connection, not delivering any more mail to the recipient MTA.

In enforced mode, however, sending MTAs MUST first check for a new `_authenticated_` policy before actually treating a message failure as fatal.

Thus the control flow for a sending MTA that does online policy application consists of the following steps:

1. Check for cached non-expired policy. If none exists, fetch the latest and cache it.
2. Validate recipient MTA against policy. If valid, deliver mail.
3. If policy invalid and policy specifies reporting, generate report.
4. If policy invalid and policy specifies rejection, perform the following steps:
 - * Check for a new (non-cached) `_authenticated_` policy. If one exists, update the current policy and go to step 1.
 - * If none exists or the newly fetched policy also fails, treat the delivery as a failure.

Understanding the details of step 4 is critical to understanding the behavior of the system as a whole.

Remember that each policy has an expiration time (which SHOULD be long, on the order of days or months) and a validation method. With these two mechanisms and the procedure specified in step 4, recipients who publish a policy have, in effect, a means of updating a cached policy at arbitrary intervals, without the risks (of a man-in-the-middle attack) they would incur if they were to shorten the policy expiration time.

4. Failure Reporting

Aggregate statistics on policy failures MAY be reported to the URI indicated in the "rua" field of the policy. SMTP STS reports contain information about policy failures to allow diagnosis of misconfigurations and malicious activity.

(There may also be a need for enabling more detailed "forensic" reporting during initial stages of a deployment. To address this, the authors consider the possibility of an optional additional "forensic reporting mode" in which more details--such as certificate chains and MTA banners--may be reported. See the section `_Future_Work_` for more details.)

Aggregate reports contain the following fields:

- o The SMTP STS policy applied (as a string)
- o The beginning and end of the reporting period

Repeated records contain the following fields:

- o Failure type: This list will start with the minimal set below, and is expected to grow over time based on real-world experience. The initial set is:
 - * `mx-mismatch`: This indicates that the MX resolved for the recipient domain did not match the MX constraint specified in the policy.
 - * `certificate-mismatch`: This indicates that the certificate presented by the receiving MX did not match the MX hostname
 - * `invalid-certificate`: This indicates that the certificate presented by the receiving MX did not validate according to the policy validation constraint. (Either it was not signed by a trusted CA or did not match the DANE TLSA record for the recipient MX.)
 - * `expired-certificate`: This indicates that the certificate has expired.
 - * `starttls-not-supported`: This indicates that the recipient MX did not support STARTTLS.
 - * `tlsa-invalid`: This indicates a validation error for Policy Domain specifying "tlsa" validation.
 - * `dnssec-invalid`: This indicates a failure to validate DNS records for a Policy Domain specifying "tlsa" validation or "dnssec" authentication.
 - * `sender-does-not-support-validation-method`: This indicates the sending system can never validate using the requested validation mechanism.

- o Count: The number of times the error was encountered.
- o Hostname: The hostname of the recipient MX.

Note that the failure types are non-exclusive; an aggregate report MAY contain overlapping counts of failure types where a single send attempt encountered multiple errors.

When sending failure reports, sending MTAs MUST NOT honor SMTP STS or DANE TLSA failures.

5. IANA Considerations

The ".well-known" URI for Policy Domains to host their STS Policies will be registered by following the procedure documented in [RFC5785] (i.e. sending a request to the "wellknown-uri-review@ietf.org" mailing list for review and comment). The proposed URI-suffix is "smtp-sts".

6. Security Considerations

SMTP Strict Transport Security protects against an active attacker who wishes to intercept or tamper with mail between hosts who support STARTTLS. There are two classes of attacks considered:

- o Foiling TLS negotiation, for example by deleting the "250 STARTTLS" response from a server or altering TLS session negotiation. This would result in the SMTP session occurring over plaintext, despite both parties supporting TLS.
- o Impersonating the destination mail server, whereby the sender might deliver the message to an impostor, who could then monitor and/or modify messages despite opportunistic TLS. This impersonation could be accomplished by spoofing the DNS MX record for the recipient domain, or by redirecting client connections to the legitimate recipient server (for example, by altering BGP routing tables).

SMTP Strict Transport Security relies on certificate validation via either TLS identity checking [RFC6125] or DANE TLSA [RFC7672]. Attackers who are able to obtain a valid certificate for the targeted recipient mail service (e.g. by compromising a certificate authority) are thus out of scope of this threat model.

In the WebPKI constraint mode, an attacker who is able to block DNS responses can suppress the delivery of an STS Policy, making the Policy Domain appear not to have an STS Policy. The caching model described in `_Policy_` `_Expirations_` is designed to resist this

attack, and there is discussion in the `_Future_ _Work_` section around future distribution mechanisms that are robust against this attack.

7. Future Work

The authors would like to suggest multiple considerations for future discussion.

- o Certificate pinning: One potential improvement in the robustness of the certificate validation methods discussed would be the deployment of public-key pinning as defined for HTTP in [RFC7469]. A policy extension supporting these semantics would enable Policy Domains to specify certificates that MUST appear in the MX certificate chain, thus providing resistance against compromised CA or DNSSEC zone keys.
- o Policy distribution: As with Certificate Transparency ([RFC6962]), it may be possible to provide a verifiable log of policy `_observations_` (meaning which policies have been observed for a given Policy Domain). This would provide insight into policy spoofing or faked policy non-existence. This may be particularly useful for Policy Domains not using DNSSEC, since it would provide sending MTAs an authoritative source for whether a policy is expected for a given domain.
- o Receive-from restrictions: Policy publishers may wish to also indicate to domains `_receiving_` mail from the Policy Domain that all such mail is expected to be sent via TLS. This may allow policy publishers to receive reports indicating sending MTA misconfigurations. However, the security properties of a "receiver-enforced" system differ from those of the current design; in particular, an active man-in-the-middle attacker may be able to exploit misconfigured sending MTAs in a way that would not be possible today with a sender-enforced model.
- o Cipher and TLS version restrictions: Policy publishers may also wish to restrict TLS negotiation to specific ciphers or TLS versions.

In addition, the authors leave currently open the following details:

- o Whether and how more detailed "forensic reporting" should be accomplished, as discussed in the section `_Failure_ _Reporting_`.
- o The registration of the `.well-known/smtp-sts` URI as per [RFC5785].

8. Appendix 1: Validation Pseudocode

```
policy = policy_from_cache()
if not policy or is_expired(policy):
    policy = policy_from_dns() // fetch and authenticate!
    update_cache = true
if policy:
    if invalid_mx_or_tls(policy): // check MX and TLS cert
        if rua:
            generate_report()
        if p_reject():
            policy = policy_from_dns() // fetch and authenticate #2!
            update_cache = true
            if invalid_mx_or_tls(policy):
                reject_message()
                update_cache = false
    if update_cache:
        cache(policy)
```

9. Appendix 2: Domain Owner STS example record

The owner wishes to begin using STS
with a policy that will solicit aggregate feedback from receivers
without affecting how the messages are processed, in order to:

- * Confirm that its legitimate messages are sent over TLS
- * Verify the validity of the certificates
- * Verify what cyphers are in use
- * Determine how many messages would be affected by a strict policy

```
_smtp_sts IN TXT ( "v=STS1; to=false; "  
                  "rua=mailto:sts-feedback@example.com " )
```

10. Appendix 3: XML Schema for Failure Reports

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.example.org/smime-sts-xml/0.1"
  xmlns:tns="http://www.example.org/smime-sts-xml/0.1">
  <!-- The time range in UTC covered by messages in this report,
        specified in seconds since epoch. -->
  <xs:complexType name="DateRangeType">
    <xs:all>
      <xs:element name="begin" type="xs:integer"/>
```

```
        <xs:element name="end" type="xs:integer"/>
    </xs:all>
</xs:complexType>

<!-- Report generator metadata. -->
<xs:complexType name="ReportMetadataType">
    <xs:sequence>
        <xs:element name="org_name" type="xs:string"/>
        <xs:element name="email" type="xs:string"/>
        <xs:element name="extra_contact_info" type="xs:string"
            minOccurs="0"/>
        <xs:element name="report_id" type="xs:string"/>
        <xs:element name="date_range" type="tns:DateRangeType"/>
    </xs:sequence>
</xs:complexType>

<!-- The constraints applied in a policy -->
<xs:simpleType name="ConstraintType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="WebPKI"/>
        <xs:enumeration value="TLSA"/>
    </xs:restriction>
</xs:simpleType>

<!-- The policy that was applied at send time. -->
<xs:complexType name="AppliedPolicyType">
    <xs:all>
        <xs:element name="domain" type="xs:string"/>
        <xs:element name="mx" type="xs:string"
            minOccurs="1" />
        <xs:element name="constraint" type="tns:ConstraintType"/>
    </xs:all>
</xs:complexType>

<!-- The possible failure types applied in a policy -->
<xs:simpleType name="FailureType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="MxMismatch"/>
        <xs:enumeration value="InvalidCertificate"/>
        <xs:enumeration value="ExpiredCertificate"/>
        <xs:enumeration value="StarttlsNotSupported"/>
        <xs:enumeration value="TlsaInvalid"/>
        <xs:enumeration value="DnssecInvalid"/>
        <xs:enumeration value="SenderDoesNotSupportValidationMethod"/>
    </xs:restriction>
</xs:simpleType>
```

```
<!-- The possible enforcement level: whether the reporter also drops
      messages -->
<xs:simpleType name="EnforcementLevelType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ReportOnly"/>
    <xs:enumeration value="Reject"/>
  </xs:restriction>
</xs:simpleType>

<!-- Record for individual failure types. -->
<xs:complexType name="FailureRecordType">
  <xs:all>
    <xs:element name="failure" type="tns:FailureType"/>
    <xs:element name="count" type="xs:integer"/>
    <xs:element name="hostname" type="xs:string"/>
    <xs:element name="connectedIp" type="xs:string" minOccurs="0"/>
    <xs:element name="sourceIp" type="xs:string" minOccurs="0"/>
  </xs:all>
</xs:complexType>

<!-- Parent -->
<xs:element name="feedback">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="version"
        type="xs:decimal"/>
      <xs:element name="report_metadata"
        type="tns:ReportMetadataType"/>
      <xs:element name="applied_policy"
        type="tns:AppliedPolicyType"/>
    </xs:sequence>
    <xs:element name="enforcement_level"
      type="tns:EnforcementLevelType"/>
    <xs:element name="record" type="tns:FailureRecordType"
      maxOccurs="unbounded"/>
  </xs:complexType>
</xs:element>
</xs:schema>
```

11. Appendix 4: Example report

```
<feedback xmlns="http://www.example.org/smtplib-sts-xml/0.1">
  <version>1</version>
  <report_metadata>
    <org_name>Company XYZ</org_name>
    <email>sts-reporting@company.com</email>
    <extra_contact_info></extra_contact_info>
    <report_id>12345</report_id>
    <date_range><begin>1439227624</begin>
    <end>1439313998</end></date_range>
  </report_metadata>
  <applied_policy>
    <domain>company.com</domain>
    <mx>*.mx.mail.company.com</mx>
    <constraint>WebPKI</constraint>
  </applied_policy>
  <enforcement_level>ReportOnly</enforcement_level>
  <record>
    <failure>ExpiredCertificate</failure>
    <count>13128</count>
    <hostname>mta7.am0.yahoodns.net.</hostname>
    <connectedIp> 98.136.216.25</connectedIp>
  </record>
  <record>
    <failure>StarttlsNotSupported</failure>
    <count>19</count>
    <hostname>mta7.am0.yahoodns.net.</hostname>
    <connectedIp>98.22.33.99</connectedIp>
  </record>
</feedback>
```

12. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, DOI 10.17487/RFC3207, February 2002, <<http://www.rfc-editor.org/info/rfc3207>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<http://www.rfc-editor.org/info/rfc3986>>.

- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<http://www.rfc-editor.org/info/rfc4033>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<http://www.rfc-editor.org/info/rfc5234>>.
- [RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", RFC 5785, DOI 10.17487/RFC5785, April 2010, <<http://www.rfc-editor.org/info/rfc5785>>.
- [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, DOI 10.17487/RFC6125, March 2011, <<http://www.rfc-editor.org/info/rfc6125>>.
- [RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, DOI 10.17487/RFC6962, June 2013, <<http://www.rfc-editor.org/info/rfc6962>>.
- [RFC7469] Evans, C., Palmer, C., and R. Sleevi, "Public Key Pinning Extension for HTTP", RFC 7469, DOI 10.17487/RFC7469, April 2015, <<http://www.rfc-editor.org/info/rfc7469>>.
- [RFC7672] Dukhovni, V. and W. Hardaker, "SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS)", RFC 7672, DOI 10.17487/RFC7672, October 2015, <<http://www.rfc-editor.org/info/rfc7672>>.

Authors' Addresses

Daniel Margolis
Google, Inc

Email: dmargolis (at) google.com

Mark Risher
Google, Inc

Email: rishe (at) google (dot com)

Nicolas Lidzborski
Google, Inc

Email: nlidz (at) google (dot com)

Wei Chuang
Google, Inc

Email: weihaw (at) google (dot com)

Brandon Long
Google, Inc

Email: blong (at) google (dot com)

Binu Ramakrishnan
Yahoo!, Inc

Email: rbinu (at) yahoo-inc (dot com)

Alexander Brotman
Comcast, Inc

Email: alexander_brotman (at) cable.comcast (dot com)

Janet Jones
Microsoft, Inc

Email: janet.jones (at) microsoft (dot com)

Franck Martin
LinkedIn

Email: fmartin (at) linkedin (dot com)

Klaus Umbach
l&l Mail & Media Development & Technology GmbH

Email: klaus.umbach (at) lundl (dot de)

Markus Laber
l&l Mail & Media Development & Technology GmbH
Email: markus.laber (at) lund1 (dot de)