1. Abstract

This document describes an extension to the SMTP service that allows an SMTP server and client to use transport-layer security to provide private, authenticated communication over the Internet. This gives SMTP agents the ability to protect some or all of their communications from eavesdroppers and attackers.

2. Introduction

SMTP [RFC-821] servers and clients normally communicate in the clear over the Internet. In many cases, this communication goes through one or more router that is not controlled or trusted by either entity. Such an untrusted router might allow a third party to monitor or alter the communications between the server and client.

Further, there is often a desire for two SMTP agents to be able to authenticate each others’ identities. For example, a secure SMTP server might only allow communications from other SMTP agents it knows, or it might act differently for messages received from an agent it knows than from one it doesn’t know.

TLS [TLS], more commonly known as SSL, is a popular mechanism for enhancing TCP communications with privacy and authentication. TLS is in wide use with the HTTP protocol, and is also being used for adding security to many other common protocols that run over TCP.

2.1 Discussion of this Draft

This draft is being discussed on the "ietf-apps-tls" mailing list.
To subscribe, send a message to:
    ietf-apps-tls-request@imc.org
with the single word
subscribe
in the body of the message. There is a Web site for the mailing list at <http://www.imc.org/ietf-apps-tls/>.

3. TLS Extension

The TLS extension to SMTP is laid out as follows:

(1) the name of the SMTP service defined here is TLS;
(2) the EHLO keyword value associated with the extension is TLS;
(3) the TLS keyword has no parameters;
(4) a new SMTP verb, "STARTTLS", is defined;
(5) no additional parameters are added to any SMTP command.

4. The TLS Keyword

The TLS keyword is used to tell the SMTP client that the SMTP server allows use of TLS. It no parameters.

5. The STARTTLS Command

The format for the STARTTLS command is:

STARTTLS

with no parameters.

After the client gives the STARTTLS command, the server responds with one of the following reply codes:

220 Ready to start TLS
501 Syntax error (no parameters allowed)
454 TLS not available due to temporary reason

A publicly-referenced server SHOULD be able to accept other SMTP commands before receiving a STARTTLS command. After receiving a 220 response to a STARTTLS command, the client SHOULD issue a STARTTLS command before giving any other SMTP commands.

A SMTP server that is not publicly referenced may choose to require that the client perform a TLS negotiation before accepting any commands. In this case, the server SHOULD return the reply code:

505 Must issue a STARTTLS command first

to every command other than STARTTLS or QUIT. If the client and server are using the ENHANCEDSTATUSCODES ESMTP extension [RFC-2034], the status code to be returned SHOULD be 5.7.0.

If the SMTP client is using pipelining as defined in RFC 1854, the STARTTLS command must be the last command in a group.

5.1 Result of the STARTTLS Command

After the TLS handshake has been completed, both parties MUST
immediately decide whether or not to continue based on the authentication and privacy achieved. The SMTP client and server may decide to move ahead even if the TLS negotiation ended with no authentication and/or no privacy because most SMTP services are performed with no authentication and no privacy, but some SMTP clients or servers may want to continue only if a particular level of authentication and/or privacy was achieved.

If the SMTP client decides that the level of authentication or privacy is not high enough for it to continue, it SHOULD issue an SMTP QUIT command immediately after the TLS negotiation is complete. If the SMTP server decides that the level of authentication or privacy is not high enough for it to continue, it SHOULD reply to every SMTP command from the client (other than a QUIT command) with the 554 reply code (with a possible text string such as "Command refused due to lack of security").

The decision of whether or not to believe the authenticity of the other party in a TLS negotiation is a local matter. However, some general rules for the decisions are:
- A SMTP client would probably only want to authenticate an SMTP server whose server certificate has a domain name that is the domain name that the client thought it was connecting to.
- A publicly-referenced SMTP server would probably want to accept any certificate from an SMTP client, and would possibly want to put distinguishing information about the certificate in the Received header of messages that were relayed or submitted from the client.

6. Usage Example

The following dialog illustrates how a client and server can start a TLS session:

S: <waits for connection on TCP port 25>
C: <opens connection>
S: 220 mail.imc.org SMTP service ready
C: EHLO mail.ietf.org
S: 250-mail.imc.org offers a warm hug of welcome
S: 250 TLS
C: STARTTLS
S: 220 Go ahead
C: <starts TLS negotiation>
C & S: <negotiate a TLS session>
C & S: <check result of negotiation>
C: <continues by sending an SMTP command>

7. Security Considerations

It should be noted that SMTP is not an end-to-end mechanism. Thus, if an SMTP client/server pair decide to add TLS privacy, they are not securing the transport from the originating mail user agent to the recipient. Further, because delivery of a single piece of mail may go between more than two SMTP servers, adding TLS privacy to one pair of servers does not mean that the entire SMTP chain has been made private. Further, just because an SMTP server can authenticate an SMTP client, it does not mean that the mail from the SMTP client was authenticated by the SMTP client when the client received it.
Both the SMTP client and server must check the result of the TLS negotiation to see whether acceptable authentication or privacy was achieved. Ignoring this step completely invalidates using TLS for security. The decision about whether acceptable authentication or privacy was achieved is made locally, is implementation-dependant, and is beyond the scope of this document.

The SMTP client and server should note carefully the result of the TLS negotiation. If the negotiation results in no privacy, or if it results in privacy using algorithms or key lengths that are deemed not strong enough, or if the authentication is not good enough for either party, the client may choose to end the SMTP session with an immediate QUIT command, or the server may choose to not accept any more SMTP commands.

A server announcing in an EHLO response that it uses a particular TLS protocol should not pose any security issues, since any use of TLS will be at least as secure as no use of TLS.

A man-in-the-middle attack can be launched by deleting the "250 TLS" response from the server. This would cause the client not to try to start a TLS session.

Another draft, [SMTP-AUTH], proposes a different mechanism that can also add privacy and security to SMTP. [SMTP-AUTH] does not allow for using TLS, but instead describes how to enable other security protocols when using SMTP.

A. References

[RFC-1869] "SMTP Service Extensions", RFC 1869
[SMTP-AUTH] "SMTP Service Extension for Authentication", draft-myers-smtp-auth

B. Revision History

***Changes from -03 to -04:

Minor grammar changes.

Expanded the wording in the third paragraph of section 7.

Added appendix C.

C. Revocation of smtps Port

An IANA port registration was made for an "smtps" port for use as a TLS-negotiated SMTP port. The email community has reached rough consensus that widespread use of such a port will be harmful to the performance,
interoperability and security of SMTP. This document hereby revokes the IANA registration of the "smtps" port and forbids future registration of a port for any "secure SMTP" service. IANA is directed to replace the port registration with an indication that the port registration was revoked, including the effective date. Two years after the effective date of revocation, the port may be re-registered for a different purpose.

D. Author’s Address

Paul Hoffman
Internet Mail Consortium
127 Segre Place
Santa Cruz, CA  95060
(408) 426-9827
phoffman@imc.org