Network Working Group   C. Newman
Internet Draft: Using TLS with IMAP4, POP3 and ACAP   Innosoft
Document: draft-newman-tls-imappop-03.txt   March 1998

Using TLS with IMAP4, POP3 and ACAP

Status of this memo

Introduction

The TLS protocol [TLS] (formerly known as SSL) provides a way to secure a connection from tampering and evesdropping. Obviously, the option of using such security is desirable for IMAP [IMAP4], POP [POP3] and ACAP [ACAP]. Although advanced SASL [SASL] authentication mechanisms can provide a lightweight version of this service, TLS is a full service security layer and is also useful in combination with plaintext password logins and other simple mechanisms as it doesn't require a site to upgrade its authentication database.

This specification defines extensions to IMAP4, ACAP and POP3 which activate TLS. It also defines a set of server security policy response codes for use with IMAP4. The response codes MAY be used independently of the TLS extension. Finally, this defines a simple PLAIN SASL mechanism for use underneath strong TLS encryption with ACAP.

[NOTE: Public discussion of this mechanism may take place on the ietf-apps-tls@imc.org mailing list with a subscription address of ietf-apps-tls-request@imc.org. Private comments may be sent to the author].

1. Conventions Used in this Document

The key words "REQUIRED", "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [KEYWORDS].

Formal syntax is defined using ABNF [ABNF].

In examples, "C:" and "S:" indicate lines sent by the client and server respectively.

2. Cipher Suite Requirements

This application profile of TLS follows the standard "Mandatory Cipher Suites" requirement as documented in the TLS specification [TLS]. Implementations MUST NOT assume any other cipher suites are present.

3. IMAP4 STARTTLS extension

When the TLS extension is present in IMAP4, "STARTTLS" is listed as a capability in response to the CAPABILITY command. This extension adds a single command, "STARTTLS" to the IMAP4 protocol which is used to begin a TLS negotiation.

3.1. STARTTLS Command

Arguments:	none
Responses:	no specific responses for this command
Result:	OK - begin TLS negotiation
	NO - security layer already active
	BAD - command unknown or arguments invalid

A TLS negotiation begins immediately after the CRLF at the end of the tagged OK response from the server. The STARTTLS command MAY be used in any state. However, a NO response MAY result if a security layer is already active. Once a client issues a STARTTLS

command, it MUST NOT issue further commands until a server response is seen.

If STARTTLS is issued in non-authenticated state, the server remains in non-authenticated state, even if client credentials are supplied during the TLS negotiation. The SASL [SASL] EXTERNAL mechanism MAY be used to authenticate once TLS client credentials are successfully exchanged, but servers supporting the STARTTLS command are not required to support the EXTERNAL mechanism.

The formal syntax for IMAP4 is amended as follows:

```
  command_any   =/  "STARTTLS"
```

```
Example:    C: a001 CAPABILITY
            S: * CAPABILITY IMAP4rev1 STARTTLS
            S: a001 OK CAPABILITY completed
            C: a002 STARTTLS
            S: a002 OK Begin TLS negotiation now
            <TLS negotation, further commands are under TLS layer>
            C: a003 LOGIN joe password
            S: a003 OK LOGIN completed
```

4. POP3 STLS extension

The POP3 STLS extension adds the STLS command to POP3 servers. If this is implemented, the POP3 extension mechanism [POP3EXT] MUST also be implemented to avoid the need for client probing of multiple commands. The capability name "STLS" indicates this command is present.

STLS

Arguments: none

Restrictions:
MAY be given in any state, but MAY fail if a security layer is already active.

Discussion:
A TLS negotiation begins immediately after the CRLF at the end of the +OK response from the server. A -ERR response MAY result if a security layer is already active. Once a client issues a STLS command, it MUST NOT issue further commands until a server response is seen.

If STLS is issued in authorization state, the server remains in authorization state, even if client credentials

are supplied during the TLS negotiation. The AUTH command [POP3-AUTH] with the EXTERNAL mechanism [SASL] MAY be used to authenticate once TLS client credentials are successfully exchanged, but servers supporting the STLS command are not required to support the EXTERNAL mechanism.

Possible Responses:
    +OK -ERR

Examples:

```
    C: STLS
    S: +OK Begin TLS negotiation
    <TLS negotiation, further commands are under TLS layer>
      ...
    C: STLS
    S: -ERR Security Layer already active
```

5. ACAP STARTTLS extension

When the TLS extension is present in ACAP, "STARTTLS" is listed as a capability in the ACAP greeting. No arguments to this capability are defined at this time. This extension adds a single command, "STARTTLS" to the ACAP protocol which is used to begin a TLS negotiation.

5.1. STARTTLS Command

```
Arguments:  none

Responses:  no specific responses for this command

Result:     OK - begin TLS negotiation
            NO - security layer already active
            BAD - command unknown or arguments invalid
```

A TLS negotiation begins immediately after the CRLF at the end of the tagged OK response from the server. The STARTTLS command MAY be used in any state. However, a NO response MAY result if a security layer is already active. Once a client issues a STARTTLS command, it MUST NOT issue further commands until a server response is seen.

If STARTTLS is issued in non-authenticated state, the server remains in non-authenticated state, even if client credentials are supplied during the TLS negotiation. The SASL [SASL] EXTERNAL mechanism MAY be used to authenticate once TLS client credentials are successfully exchanged, but servers supporting the STARTTLS

command are not required to support the EXTERNAL mechanism.

The formal syntax for ACAP is amended as follows:

```
  command_any   =/  "STARTTLS"
```

```
Example:    S: * ACAP (SASL "CRAM-MD5" "PLAIN" "EXTERNAL")
               (STARTTLS)
            C: a002 STARTTLS
            S: a002 OK "Begin TLS negotiation now"
            <TLS negotiation, further commands are under TLS layer>
```

6. PLAIN SASL mechanism

Plaintext passwords are simple, interoperate with almost all existing operating system authentication databases, and are useful for a smooth transition to a more secure password-based authentication mechanism. The drawback is that they are unacceptable for use unencrypted over the network.

This defines a PLAIN SASL mechanism for use with ACAP and future protocols with no plaintext login command. This MUST NOT be implemented unless TLS (or an equivalent security layer) is also implemented.

The SASL [SASL] mechanism name is "PLAIN".

The mechanism consists of a single message from the client to the server. The client sends the authorization identity (identity to login as), followed by a US-ASCII NUL character, followed by the authentication identity (identity whose password will be used), followed by a US-ASCII NUL character, followed by the plaintext password. The client may leave the authorization identity empty to indicate that it is the same as the authentication identity.

The server will verify the authentication identity and password with the system authentication database and verify that the authentication credentials permit the client to login as the authorization identity. If both steps succeed, the user is logged in.

The server MAY also use the password to initialize any new authentication database, such as one suitable for CRAM-MD5 [CRAM-MD5], ACAP's mandatory to implement authentication mechanism.

Non-US-ASCII characters are permitted as long as they are represented in UTF-8 [UTF-8]. Use of non-visible characters or characters which a user may be unable to enter on some keyboards is

discouraged.

The formal grammar for the client message using Augmented BNF [ABNF] follows.

```
message         = [authorize-id] NUL authenticate-id NUL password

authenticate-id = 1*UTF8-SAFE
                  ; MUST accept up to 255 octets

authorize-id    = 1*UTF8-SAFE
                  ; MUST accept up to 255 octets

password        = *UTF8-SAFE
                  ; MUST accept passwords up to 255 octets

NUL             = %x00

UTF8-SAFE       = %x01-09 / %x0B-0C / %x0E-7F / UTF8-2 /
                  UTF8-3 / UTF8-4 / UTF8-5 / UTF8-6

UTF8-1          = %x80-BF

UTF8-2          = %xC0-DF UTF8-1

UTF8-3          = %xE0-EF 2UTF8-1

UTF8-4          = %xF0-F7 3UTF8-1

UTF8-5          = %xF8-FB 4UTF8-1

UTF8-6          = %xFC-FD 5UTF8-1
```

Here is an example of how this might be used to initialize a CRAM-MD5 authentication database for ACAP:

```
Example:    S: * ACAP (SASL "CRAM-MD5" "PLAIN" "EXTERNAL")
               (STARTTLS)
            C: a001 AUTHENTICATE "CRAM-MD5"
            S: + "<1896.697170952@postoffice.reston.mci.net>"
            C: "tim b913a602c7eda7a495b4e6e7334d3890"
            S: a001 NO (TRANSITION-NEEDED)
               "Please change your password, or use TLS to login"
            C: a002 STARTTLS
            S: a002 OK "Begin TLS negotiation now"
            <TLS negotation, further commands are under TLS layer>
            C: a003 AUTHENTICATE "PLAIN" {21+}
            C: <NUL>tim<NUL>tanstaaftanstaaf
```

```
            S: a003 OK AUTHENTICATE completed
```

Note: In this example, <NUL> represents a single ASCII NUL octet.

Here is an example session where a client erroneously attempts to use PLAIN prior to starting TLS:

```
Example:    S: * ACAP (SASL "CRAM-MD5" "PLAIN" "EXTERNAL")
               (STARTTLS)
            C: a001 AUTHENTICATE "PLAIN" {21}
            S: a001 NO (ENCRYPT-NEEDED)
               "Can't use PLAIN without encryption"
```

7. imaps and pop3s ports

The common practice of using a separate port for a secure version of each protocol has a number of disadvantages in the IMAP [IMAP4], ACAP [ACAP] and POP [POP3] environment. Rather than using the best security available, it means that clients have to be explicitly configured to use the separate secure port or suffer the performance loss of probing for active ports. For IMAP and ACAP, this is even more serious as it would require a new URL scheme which could only be resolved by TLS-enabled clients.

Separate "imaps" and "pop3s" ports were registered for use with TLS. Use of these ports is discouraged in favor of the STARTTLS or STLS command.

One of the arguments used in favor of the separate port technique is that it simplifies configuration of firewalls which filter by IP port. However, a quality server implementation running on the standard port can be configured to require use of the STARTTLS command or a suitably strong SASL mechanism for non-local connections. This provides superior functionality as the client need not be re-configured for use outside the firewall and faster non-plaintext SASL mechanisms may be acceptable to many sites for non-local connections.

8. Security Considerations

The mechanisms described in this document only apply to protecting a single connection. Messages transferred over IMAP or POP3 are still available to server administrators and usually subject to evesdropping, tampering and forgery when transmitted through SMTP or NNTP. Protecting messages requires an object security mechanism such as PGP MIME [PGP-MIME].

An active attacker can remove STARTTLS from the capability list. In order to detect such an attack, clients SHOULD either warn the user when session protection is not active, or be configurable to refuse to proceed without an acceptable level of security.

An active attacker can always cause a down-negotiation to the weakest authentication mechanism or cipher suite available. For this reason, implementations need to be configurable to refuse weak mechanisms or cipher suites.

When the PLAIN mechanism is used with TLS, the server gains the ability to impersonate the user to all services with the same password. The PLAIN mechanism MUST NOT be used without an active encryption layer using a key with an effective key length greater than 56 bits, otherwise a passive attacker can gain the ability to impersonate the user.

9. References

[ABNF] Crocker, Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium, Demon Internet Ltd, November 1997.

[ACAP] Newman, Myers, "ACAP -- Application Configuration Access Protocol", RFC 2244, Innosoft, Netscape, November 1997.

[CRAM-MD5] Klensin, Catoe, Krumviede, "IMAP/POP AUTHorize Extension for Simple Challenge/Response", RFC 2195, MCI, September 1997.

[IMAP4] Crispin, M., "Internet Message Access Protocol - Version 4rev1", RFC 2060, University of Washington, December 1996.

[IMAP-AUTH] Myers, J., "IMAP4 Authentication Mechanism", RFC 1731, Carnegie-Mellon University, December 1994.

[KEYWORDS] Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, Harvard University, March 1997.

[PGP-MIME] Elkins, M., "MIME Security with Pretty Good Privacy (PGP)", RFC 2015, The Aerospace Corporation, October 1996.

[POP3] Myers, J., Rose, M., "Post Office Protocol - Version 3", RFC 1939, Carnegie Mellon, Dover Beach Consulting, Inc., May 1996.

[POP3EXT] Gellens, Newman, Lundblade "POP3 Extension Mechanism", Work in progress.

[POP-AUTH] Myers, "POP3 AUTHentication command", RFC 1734, Carnegie Mellon, December 1994.

[SASL] Myers, "Simple Authentication and Security Layer (SASL)", RFC 2222, Netscape Communications, October 1997.

[TLS] Dierks, Allen, "The TLS Protocol Version 1.0", Work in progress.

[UTF-8] Yergeau, F. "UTF-8, a transformation format of ISO 10646", RFC 2279, Alis Technologies, January 1998.

10. Author's Address

Chris Newman
Innosoft International, Inc.
1050 Lakes Drive
West Covina, CA 91790 USA

Email: chris.newman@innosoft.com