Independent Submission
Internet-Draft
Intended status: Standards Track
Expires: July 8, 2017

J. Elie
January 4, 2017

Modernization of RFC 4642
draft-elie-nntp-tls-recommendations-rfc4642bis-01

Abstract

This document shows the sections that changed between RFC 4642 and draft-elie-nntp-tls-recommendations. The -00 version contains the wording in RFC 4642. The -01 version contains the wording in draft-elie-nntp-tls-recommendations-04.

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 July 8, 2017.

1. Wording updated by draft-elie-nntp-tls-recommendations-04

Abstract

This memo defines an extension to the Network News Transfer Protocol (NNTP) that allows an NNTP client and server to use Transport Layer Security (TLS). The primary goal is to provide encryption for single-link confidentiality purposes, but data integrity, and (optional) certificate-based peer entity authentication are also possible.

1. Introduction

Historically, unencrypted NNTP [NNTP] connections were satisfactory for most purposes. However, sending passwords unencrypted over the network is no longer appropriate, and sometimes integrity and/or confidentiality protection are desired for the entire connection.

The TLS protocol (formerly known as SSL) provides a way to secure an application protocol from tampering and eavesdropping. Although advanced SASL authentication mechanisms [NNTP-AUTH] can provide a lightweight version of this service, TLS is complimentary to both simple authentication-only SASL mechanisms and deployed clear-text password login commands.

TCP port 563 is dedicated to NNTP over TLS, and registered in the IANA Service Name and Transport Protocol Port Number Registry for that usage. NNTP implementations using TCP port 563 begin the TLS negotiation immediately upon connection and then continue with the initial steps of an NNTP session. This immediate TLS negotiation on a separate port (referred to in this document as "implicit TLS") is the preferred way of using TLS with NNTP.

If a host wishes to offer separate servers for transit and reading clients (Section 3.4.1 of [NNTP]), TCP port 563 SHOULD be used for implicit TLS with the reading server, and an unused port of its choice different than TCP port 433 SHOULD be used for implicit TLS with the transit server. The ports used for implicit TLS should be clearly communicated to the clients, and specifically that no plain-text communication occurs before the TLS session is negotiated.

As some existing implementations negotiate TLS via a dynamic upgrade from unencrypted to TLS-protected traffic during an NNTP session on well-known TCP ports 119 or 433, this specification formalizes the STARTTLS command in use for that purpose. However, as already mentioned above, implementations SHOULD use implicit TLS on a separate port.

Note: a common alternative to protect NNTP exchanges with transit servers that do not implement TLS is the use of IPsec with encryption [RFC4301].

2.2.2. Description

A client issues the STARTTLS command to request negotiation of TLS. The STARTTLS command is usually used to initiate session security, although it can also be used for client and/or server certificate authentication.

An NNTP server returns the 483 response to indicate that a secure or encrypted connection is required for the command sent by the client. Use of the STARTTLS command as described below is one way to establish a connection with these properties. The client MAY therefore use the STARTTLS command after receiving a 483 response.

If a server advertises the STARTTLS capability, a client MAY attempt to use the STARTTLS command at any time during a session to negotiate TLS without having received a 483 response. Servers SHOULD accept such unsolicited TLS negotiation requests.

If the server is unable to initiate the TLS negotiation for any reason (e.g., a server configuration or resource problem), the server MUST reject the STARTTLS command with a 580 response. Then, it SHOULD either reject subsequent restricted NNTP commands from the client with a 483 response code (possibly with a text string such as "Command refused due to lack of security") or reject a subsequent restricted command with a 400 response code (possibly with a text string such as "Connection closing due to lack of security") and close the connection. Otherwise, the server issues a 382 response, and TLS negotiation begins. A server MUST NOT under any circumstances reply to a STARTTLS command with either a 480 or 483 response.

If the client receives a failure response to STARTTLS, the client must decide whether or not to continue the NNTP session. Such a decision is based on local policy. For instance, if TLS was being used for client authentication, the client might try to continue the session in case the server allows it to do so even with no authentication. However, if TLS was being negotiated for encryption, a client that gets a failure response needs to decide whether to continue without TLS encryption, to wait and try again later, or to give up and notify the user of the error.

Upon receiving a 382 response to a STARTTLS command, the client MUST start the TLS negotiation before giving any other NNTP commands. The TLS negotiation begins for both the client and server with the first

octet following the CRLF of the 382 response. If, after having issued the STARTTLS command, the client finds out that some failure prevents it from actually starting a TLS handshake, then it SHOULD immediately close the connection.

If the TLS negotiation fails, both client and server SHOULD immediately close the connection. Note that while continuing the NNTP session is theoretically possible, in practice a TLS negotiation failure often leaves the session in an indeterminate state; therefore, interoperability can not be guaranteed.

Upon successful completion of the TLS handshake, the NNTP protocol is reset to the state immediately after the initial greeting response (see 5.1 of [NNTP]) has been sent, with the exception that if a MODE READER command has been issued, its effects (if any) are not reversed. At this point, as no greeting is sent, the next step is for the client to send a command. The server MUST discard any knowledge obtained from the client, such as the current newsgroup and article number, that was not obtained from the TLS negotiation itself. Likewise, the client SHOULD discard and MUST NOT rely on any knowledge obtained from the server, such as the capability list, which was not obtained from the TLS negotiation itself.

The server remains in the non-authenticated state, even if client credentials are supplied during the TLS negotiation. The AUTHINFO SASL command [NNTP-AUTH] with the EXTERNAL mechanism [SASL] MAY be used to authenticate once TLS client credentials are successfully exchanged, but servers supporting the STARTTLS command are not required to support AUTHINFO in general or the EXTERNAL mechanism in particular. The server MAY use information from the client certificate for identification of connections or posted articles (either in its logs or directly in posted articles).

Both the client and the server MUST know if there is a TLS session active. A client MUST NOT attempt to start a TLS session if a TLS session is already active. A server MUST NOT return the STARTTLS capability label in response to a CAPABILITIES command received after a TLS handshake has completed, and a server MUST respond with a 502 response code if a STARTTLS command is received while a TLS session is already active. Additionally, the client MUST NOT issue a MODE READER command while a TLS session is active, and a server MUST NOT advertise the MODE-READER capability.

The capability list returned in response to a CAPABILITIES command received after a successful TLS handshake MAY be different from the list returned before the TLS handshake. For example, an NNTP server supporting SASL [NNTP-AUTH] might not want to advertise support for a particular mechanism unless a client has sent an appropriate client

certificate during a TLS handshake.

5. Security Considerations

Security issues are discussed throughout this memo.

In general, the security considerations of the TLS protocol [TLS] and any implemented extensions [TLS-EXT] are applicable here; only the most important are highlighted specifically below. Also, this extension is not intended to cure the security considerations described in Section 12 of [NNTP]; those considerations remain relevant to any NNTP implementation.

Before the TLS handshake has begun, any protocol interactions are performed in the clear and may be modified by an active attacker. For this reason, clients and servers MUST discard any sensitive knowledge obtained prior to the start of the TLS handshake upon the establishment of a security layer. Furthermore, the CAPABILITIES command SHOULD be re-issued upon the establishment of a security layer, and other protocol state SHOULD be re-negotiated as well.

Note that NNTP is not an end-to-end mechanism. Thus, if an NNTP client/server pair decide to add TLS confidentiality, they are securing the transport only for that link. Similarly, because delivery of a single Netnews article may go between more than two NNTP servers, adding TLS confidentiality to one pair of servers does not mean that the entire NNTP chain has been made private. Furthermore, just because an NNTP server can authenticate an NNTP client, it does not mean that the articles from the NNTP client were authenticated by the NNTP client when the client itself received them (prior to forwarding them to the server).

During TLS negotiation, the client MUST verify the server's identity in order to prevent man-in-the-middle attacks. The client MUST follow the rules and guidelines defined in [RFC6125], where the reference identifier MUST be the server hostname that the client used to open the connection, and that is also specified in the TLS "server_name" extension [RFC6066]. The following NNTP-specific consideration applies: DNS domain names in server certificates MAY contain the wildcard character "*" as the complete left-most label within the identifier.

If the match fails, the client MUST follow the recommendations in Section 6.6 of [RFC6125] regarding certificate pinning and fallback.

Beyond server identity checking, clients also MUST apply the procedures specified in [RFC5280] for general certificate

validation (e.g., certificate integrity, signing, and path validation).

A man-in-the-middle attack can be launched by deleting the STARTTLS capability label in the CAPABILITIES response from the server. This would cause the client not to try to start a TLS session. Another man-in-the-middle attack would allow the server to announce its STARTTLS capability, but alter the client's request to start TLS and the server's response. An NNTP client can partially protect against these attacks by recording the fact that a particular NNTP server offers TLS during one session and generating an alarm if it does not appear in the CAPABILITIES response for a later session. (Of course, the STARTTLS capability would not be listed after a security layer is in place.)

If the client receives a 483 or 580 response, the client has to decide what to do next. The client has to choose among three main options: to go ahead with the rest of the NNTP session, to (re)try TLS later in the session, or to give up and postpone newsreading/transport activity. If an error occurs, the client can assume that the server may be able to negotiate TLS in the future and should try to negotiate TLS in a later session. However, if the client and server were only using TLS for authentication and no previous 480 response was received, the client may want to proceed with the NNTP session, in case some of the operations the client wanted to perform are accepted by the server even if the client is unauthenticated.

7.1. Normative References

[ABNF] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[NNTP] Feather, C., "Network News Transfer Protocol (NNTP)", RFC 3977, October 2006.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, March 2011.

[TLS] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[TLS-EXT] Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., and T. Wright, "Transport Layer Security (TLS) Extensions", RFC 4366, April 2006.

7.2. Informative References

[NNTP-AUTH] Vinocur, J., Murchison, K., and C. Newman, "Network News Transfer Protocol (NNTP) Extension for Authentication", RFC 4643, October 2006.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[SASL] Melninov, A., Ed. and K. Zeilenga, Ed, "Simple Authentication and Security Layer (SASL)", RFC 4422, June 2006.

[TLS-IMAPPOP] Newman, C., "Using TLS with IMAP, POP3 and ACAP", RFC 2595, June 1999.

Author's Address

Julien Elie
10 allee Clovis
Noisy-le-Grand 93160
France

EMail: julien@trigofacile.com
URI: http://www.trigofacile.com/