

Telnet Authentication and Encryption: GSSAPI Option

Status of this Memo

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are 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''.

To learn the current status of any Internet Draft, please check the `1id-abstracts.txt` listing contained in one of the Internet Drafts Shadow Directories on `ds.internic.net` (US East Coast), `venera.isi.edu` (US West Coast), `munari.oz.au` (Pacific Rim), or `nic.nordu.net` (Europe).

Abstract

The Telnet Authentication and Encryption Option provides a framework for the passing of authentication information through the TELNET session. This document specifies how the Generic Security Service Application Program Interface (GSSAPI) uses the framework to establish the security services of authentication and integrity for a Telnet session.

1. Introduction

The Telnet Authentication and Encryption Option provides a framework for the passing of authentication information through the TELNET session and provides a mechanism to enable encryption of the data stream as a side effect of successful authentication. This document specifies how the Generic Security Service Application Program Interface (GSSAPI) [[1](#)] uses the framework to establish the security services of authentication and integrity for a Telnet session. No

INTERNET-DRAFT

Telnet Security: GSSAPI Option

July 1995

specification is included for the enabling of encryption. Quoting from the GSSAPI specification:

The GSS-API per-message protection service primitives, as the category name implies, are oriented to operation at the granularity of protocol data units. They perform cryptographic operations on the data units, transfer cryptographic control information in tokens, and, in the case of GSS_Wrap(), encapsulate the protected data unit. As such, these primitives are not oriented to efficient data protection for stream-paradigm protocols (e.g., Telnet) if cryptography must be applied on an octet-by-octet basis.

However, work is proceeding in this area and this document will be revised when it is complete.

This version of this specification restricts the negotiation to client to server (as would be indicated by the use of the AUTH_CLIENT_TO_SERVER modifier) and does not allow the use of any modifiers. This latter restriction is not a deficiency since the GSSAPI provides for all of the functionality supported by the modifiers.

2. Specification

This specification makes reference to a "server" and a "client". For the purposes of this document, the "server" is the side of the connection that did the passive TCP open (TCP LISTEN state), and the "client" is the side of the connection that did the active open.

To use the GSSAPI authentication type, there are two steps to be completed by the client and server: negotiate the use of GSSAPI and exchange tokens until the authentication is complete. The client begins the authentication exchange by calling GSS_Init_Sec_Context. If an output_token is returned, the token is sent to the server.

The server continues the authentication exchange by calling GSS_Accept_Sec_Context using the received token as input_token. If an output_token is returned, the token is sent to the client.

Both the client and server continue exchanging tokens as long as they

are returned by `GS_Init_Sec_Context` and `GSS_Accept_Sec_Context`, respectively. Each of these calls will return either `GSS_S_CONTINUE_NEEDED` or `GSS_S_COMPLETE` to their respective callers indicating if they should expect another token or not, respectively. If an unexpected token is received or an expected token is not received, either the client or server may terminate the connection.

The following example illustrates the use of this authentication type.

Client

Server

IAC DO AUTH_ENCRYPT

IAC WILL AUTH_ENCRYPT

[The server is now free to request authentication information.
]

IAC SB AUTH_ENCRYPT SEND GSSAPI
IAC SE

[The server has requested GSSAPI authentication. The client should begin the authentication exchange by calling `GSS_Init_Sec_Context` at this time. The `output_token` must then be sent to the server.]

IAC SB AUTH_ENCRYPT IS GSSAPI
<authentication-data> IAC SE

[The authentication-data is sent as a binary sequence of octets. The server continues the authentication exchange by calling `GSS_Accept_Sec_Context` with the received authentication-data as the `input_token`. If an `output_token` is returned, it is sent to the client.]

IAC SB AUTH_ENCRYPT REPLY GSSAPI
<authentication-data> IAC SE

[The client calls `GSS_Init_Sec_Context` using the received authentication-data as the `input_token`. If an `output_token` is returned, the exchange is continued with the client sending

another "IAC SB AUTH_ENCRYPT IS ... IAC SE" command. When both the client and the server receive GSS_S_COMPLETE return codes and the last one of them to receive authentication-data from the other is not returned an output_token, the authentication exchange is complete.]

[3.](#) Security Considerations

This document is about a negotiation to establish the use of an authentication service.

[4.](#) References

[1] J. Linn. Generic Security Service Application Program Interface, Version 2. Work in progress.

J. Galvin, et. al.

Expires January 1996

[Page 3]

INTERNET-DRAFT

Telnet Security: GSSAPI Option

July 1995

[5.](#) Authors' Address

Jim Galvin <galvin@tis.com>
Sandy Murphy <murphy@tis.com>
Dave Balenson <balenson@tis.com>

Trusted Information Systems
3060 Washington Road
Glenwood, MD 21738

Phone: 301.854.6889

[J.](#) Galvin, et. al.

Expires January 1996

[Page 4]