Paul Ford-Hutchinson
IBM UK Ltd
Tim Hudson
CryptSoft Pty Ltd
Eric Murray
Independent Security Consultant

<draft-murray-auth-ftp-ssl-00.txt>
Updates: RFC 959
INTERNET-DRAFT
This document expires on 1st June 1997
26th November, 1996

Secure FTP over SSL

Status of this Memo

Abstract

This document describes an extension to the FTP protocol [RFC-959] that allows secured, authenticated communications to take place on the control and data channels using the SSL protocol. This enables secure, authenticated file transfer across the Internet. The document describes a specific mechanism for negotiating SSL session protection in a manner that allows other encryption/authentication techniques to be proposed in a complemetary manner.

The new command 'AUTH' is proposed and requests that the FTP server will accept an SSL negotation and continue the rest of the FTP conversation (on both control and data connections) in a secure manner.

The following new commands are introduced in this specification:

AUTH (Authentication/Security Mechanism),

The following new replies are introduced in this specification:

232, 234, 334, 431, 534, 535, 536

Note that this specification is compatible with the ftp RFC [RFC-959].

1. Introduction

The File Transfer Protocol (FTP) currently defined in [RFC-959] and in place on the Internet is an excellent mechanism for exchanging files, however its lack of strong, private authentication and privacy of the data being transferred prevent its use in some situations. This document details some extensions to the existing FTP protocol that will allow the FTP Client and Server to use the SSL protocol to authenticate and secure the entire FTP session.

The SSL protocol is described in [SSL-DESC].

This document does not intend in any way to describe generic Security mechanisms for use with the FTP protocol, but is intended as a description of exactly how SSL should be incorporated into it. However, it is an intention to enable SSL in such a manner that the mechanism used (the AUTH command) may be utilised in an extensible way for other security schemes. In this way this proposal does not prevent future FTP protocol developments.

For example, this draft is compatible with the SRA authentication mechanism described in [SRA-FTP].

2. New FTP Commands

Request Authentication (AUTH)

The Request Authentication command is issued by the Client and is really a question to see if the server supports the additional command with the method specified. For all versions of SSL the method string is "SSL" (this string is case sensitive).

- If the server does not recognise the AUTH command then RFC 959 states that the correct response should be the 500 reply.

- If the server does not support the AUTH command then RFC 959 states that the correct response should be the 502 reply.

- If the server supports the AUTH command but does not support SSL then the reply should be 504.

- If the server supports the AUTH command and does support SSL but does not wish to use SSL for the connection then the reply will be the new 534 reply.

- If the server recognises but does not support the AUTH command then the correct reply should be 502.

- If the server supports the AUTH command but does not wish to honour it at this time, perhaps because only one AUTH is allowed per session, then a new reply, 536 will be issued.

- If the server supports the AUTH command but cannot perform SSL negotiation for some transient reason then the correct reply is a new reply, 431.

- If the server supports the AUTH command and can start an SSL negotiation then the correct reply will be 334, a new reply.

Upon receipt of a 334 reply from the Server, the FTP Client should start the SSL negotiation; the FTP client acting as the SSL client and the FTP server as the SSL server.

Once the SSL exchange has been completed, one of the following replies should be issued.

232 - All requisite authentication has been completed, there is no need to use USER and PASS to gain Authorization.

234 - The SSL Authentication procedure was successful, continue with session as normal.

431 - The SSL Authentication procedure failed for some transient reason.

534 - The SSL Authentication procedure failed due to Client/Server mismatch in the SSL protocol negotiation.

535 - The SSL Authentication procedure failed for some other reason

When issuing a 334 reply, the server must also clear out any internal authentication information (e.g. USER, PASS, ACCT) which was established.

3. New FTP Replies

The new reply codes are defined to reply to the new AUTH command.

3.1. New individual reply codes

232 User logged in, authorized by security data exchange.
234 Security data exchange complete.

334 Start Security data exchange.

431 Need some unavailable resource to process security.

534 Request denied for policy reasons.
535 Failed security check (hash, sequence, etc).
536 Command refused at this time

Note: As in the FTP specification [RFC-959], the text for these replies is not fixed, but is server implementation dependant.

4. Data Connection Behaviour

The Data Connection in the FTP model can be used in one of three ways. (Note: these descriptions are not necessarily correctly placed in chronological order, but do describe the steps required.)

a) Classic FTP Client/Server data exchange.
- The Client obtains a port, sends the port number to the Server, the Server connects to the Client. The Client issues a send or receive request to the Server and the data transfer commences.

b) Firewall Friendly Client/Server data exchange (as discussed in [FTP-SOCKS])
- The Client requests that the Server open a port, the Server obtains a port and returns it to the Client. The Client connects to the server on this port. The Client issues a send or receive request and the data transfer commences.

c) Client Initiated Server/Server data exchange. (proxy or PASV connections)
- The Client requests that Server A opens a port, Server A obtains a port and returns it to the Client. The Client sends this port number to Server B. Server B connects to Server A. The Client sends a send or receive request to Server A and the complement to Server B and the data transfer commences. In this model Server A is the proxy or PASV host and is a Client for the Data Connection to Server B.

For a) and b) the FTP Client will be the SSL Client and the FTP Server will be the SSL Server.

That is to say that it does not matter which side initiates the connection with a connect() call or which side reacts to the connection via the accept() call, the FTP client as defined in [RFC-959] will always be the SSL client as defined in [SSL-DESC].

In scenario c) there will be a problem in that neither Server A nor Server B will be the SSL Client given the fact that an FTP Server

must act as an SSL Server for Firewall Friendly FTP [FTP-SOCKS].

5. Considerations outside the scope of this protocol specification

Server/Server Data transfer using the PASV command as noted above is not specified in this document.

Server Policy Decisions

- Which SSL Cipher Suites to support/allow
- Which SSL versions to support/allow
- Authorisation model (is the SSL Authentication sufficient Authorisation ?)
- Authentication Hierarchy (Standard X.509 Certificate managment issues)
- When to allow the AUTH command
    - Only before USER
    - Anytime
    - Just once per session
- Other commands disallowed before AUTH has succesfully completed
    - do not accept USER (or PASS, ACCT ...) until AUTH has finished.

6. Implementing SSL without the AUTH command

An alternative SSL negotiation mechanism is to negotiate SSL upon Client connection to a well known port. Since this requires prior knowledge by the server (in order to refrain from sending the 220 message) the port number must be different from the standard ftp port. If a Client/Server implementation chooses this mechanism then the AUTH command is not required, however the discussion in this document on the behaviour of the Data connection is still valid and negotiating SSL on the Data port is mandatory.

This port number (for ftps) is being requested from the IANA.

7. Implementation Recommendations

While there are no restrictions on client and server policy, there are a few guidelines that the authors recommend following.

7.1 Client Implementations

Clients should allow a command line parameter to let the user specify that any failure to negotiate a secure session will cause the session to be dropped. (A suggestion is '-z secure')

If a client supports both AUTH and non-AUTH mechanisms, there will need to be a command line argument to instruct the client which mechanism to choose. (A suggestion is '-z connect')

7.2 Server Implementations

Servers should be configurable via a command line invocation

option to reject USER, PASS etc commands for connections that have not been secured. (A suggestion is '-z secure')

If a server supports both AUTH and non-AUTH mechanisms, there will need to be a command line argument to instruct the server which mechanism to choose. (A suggestion is '-z connect')

8. Declarative specifications

These sections are modelled after sections 5.3 and 5.4 of RFC 959, which describe the same information, except for the standard FTP commands and replies.

8.1. FTP Authentication command and argument

AUTH <SP> <mechanism-name> <CRLF>

This document only discusses behaviour when:-

<mechanism-name> ::= SSL

8.2. Command-Reply sequences

Security Association Setup

```
AUTH
   234
   334
   504, 534, 431
   500, 501, 421, 502
```

9. Security Considerations

This entire document deals with security considerations related to the File Transfer Protocol.

10. Acknowledgements

Netscape Communications Corp for the SSL protocol; Eric Young for the SSLeay libraries; M Horowitz and S J Lunt for the internet draft, "draft-ietf-cat-ftpsec-08.txt", the basis from which this has been developed; University of California, Berkley for the original implementations of ftp and ftpd on which the initial implementation of these extensions were layered.

11. References

[FTP-SOCKS] Bellovin, S., "Firewall-Friendly FTP"
RFC 1579, February 1994.

[SSL-DESC] A description of the SSL protocol.
The actual protocol version is not relevant to this draft, however the current version of SSL is described in
Freier, A., "The SSL Protocol Version 3.0"
draft-ietf-tls-ssl-version3-00.txt.

[RFC-959] Postel, J., "File Transfer Protocol"
RFC 959, October 1985.

[SRA-FTP] "SRA - Secure RPC Authentication for TELNET and FTP Version

   1.1"
      file://ftp.funet.fi/security/login/telnet/doc/sra/sra.README

12. Author's Contact Addresses

```
Paul Ford-Hutchinson        Tim Hudson              Eric Murray
  OSU 1                       CryptSoft Pty Ltd      LNE Consulting
  IBM UK Ltd                  PO Box 6324
  PO Box 31                   Fairfield 4103
  Birmingham Road             Queensland
  Warwick                     Australia
  England
  CV34 5JL

Tel - +44 (0)1926 464836      +61 7 32781581
Fax - +44 (0)1926 496482
email - pfh@uk.ibm.com          tjh@cryptsoft.com        ericm@lne.com
```