Network Working Group   D. Borman, Editor
Request for Comments: 1416   Cray Research, Inc.
Obsoletes: 1409   February 1993

Telnet Authentication Option

Status of this Memo

This memo defines an Experimental Protocol for the Internet community. Discussion and suggestions for improvement are requested. Please refer to the current edition of the "IAB Official Protocol Standards" for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Note

This RFC 1416 replaces RFC 1409, which has an important typographical error in the example on page 6 (one occurance of "REPLY" should be "IS").

1. Command Names and Codes

AUTHENTICATION	37
IS	0
SEND	1
REPLY	2
NAME	3

Authentication Types	
NULL	0
KERBEROS_V4	1
KERBEROS_V5	2
SPX	3
RSA	6
LOKI	10

Modifiers		
AUTH_WHO_MASK	1	
AUTH_CLIENT_TO_SERVER		0
AUTH_SERVER_TO_CLIENT		1
AUTH_HOW_MASK	2	
AUTH_HOW_ONE_WAY		0
AUTH_HOW_MUTUAL		2

2. Command Meanings

This document 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.

IAC WILL AUTHENTICATION

The client side of the connection sends this command to indicate that it is willing to send and receive authentication information.

IAC DO AUTHENTICATION

The servers side of the connection sends this command to indicate that it is willing to send and receive authentication information.

IAC WONT AUTHENTICATION

The client side of the connection sends this command to indicate that it refuses to send or receive authentication information; the server side sends this command if it receives a DO AUTHENTICATION command.

IAC DONT AUTHENTICATION

The server side of the connection sends this command to indicate that it refuses to send or receive authentication information; the client side sends this command if it receives a WILL AUTHENTICATION command.

IAC SB AUTHENTICATION SEND authentication-type-pair-list IAC SE

The sender of this command (the server) requests that the remote side send authentication information for one of the authentication types listed in "authentication-type-pair-list". The "authentication-type-pair-list" is an ordered list of "authentication-type" pairs. Only the server side (DO AUTHENTICATION) is allowed to send this.

IAC SB AUTHENTICATION IS authentication-type-pair <auth data> IAC SE

The sender of this command (the client) is sending the authentication information for authentication type "authentication-type-pair". Only the client side (WILL AUTHENTICATION) is allowed to send this.

IAC SB AUTHENTICATION REPLY authentication-type-pair <auth data> IAC SE

The sender of this command (the server) is sending a reply to the the authentication information received in a previous IS command. Only the server side (DO AUTHENTICATION) is allowed to send this.

IAC SB AUTHENTICATION NAME remote-user IAC SE

This optional command is sent to specify the account name on the remote host that the user wishes to be authorized to use. Note that authentication may succeed, and the authorization to use a particular account may still fail. Some authentication mechanisms may ignore this command.

The "authentication-type-pair" is two octets, the first is the authentication type (as listed in Section 1, additions to this list must be registered with the Internet Assigned Numbers Authority (IANA)), and the second is a modifier to the type. There are currently two one bit fields defined in the modifier, the AUTH_WHO_MASK bit and the AUTH_HOW_MASK bit, so there are four possible combinations:

AUTH_CLIENT_TO_SERVER
AUTH_HOW_ONE_WAY

The client will send authentication information about the local user to the server. If the negotiation is successful, the server will have authenticated the user on the client side of the connection.

AUTH_SERVER_TO_CLIENT
AUTH_HOW_ONE_WAY

The server will authenticate itself to the client. If the negotiation is successful, the client will know that it is connected to the server that it wants to be connected to.

AUTH_CLIENT_TO_SERVER
AUTH_HOW_MUTUAL

The client will send authentication information about the local user to the server, and then the server will authenticate itself to the client. If the negotiation is successful, the server will have authenticated the user on the client side of the connection, and the client will know that it is connected to the server that it wants to be connected to.

AUTH_SERVER_TO_CLIENT
AUTH_HOW_MUTUAL

The server will authenticate itself to the client, and then the client will authenticate itself to the server. If the negotiation is successful, the client will know that it is connected to the server that it wants to be connected to, and the server will know that the client is who it claims to be.

3. Default Specification

The default specification for this option is

WONT AUTHENTICATION
DONT AUTHENTICATION

meaning there will not be any exchange of authentication information.

4. Motivation

One of the deficiencies of the Telnet protocol is that in order to log into remote systems, users have to type their passwords, which are passed in clear text through the network. If the connections goes through untrusted networks, there is the possibility that passwords will be compromised by someone watching the packets as they go by.

The purpose of the AUTHENTICATION option is to provide a framework for the passing of authentication information through the TELNET session. This means that: 1) the users password will not be sent in clear text across the network, and 2) if the front end telnet process has the appropriate authentication information, it can automatically send the information, and the user will not have to type any password.

It is intended that the AUTHENTICATION option be general enough that it can be used to pass information for any authentication system.

5. Security Implications

The ability to negotiate a common authentication mechanism between client and server is a feature of the authentication option that should be used with caution. When the negotiation is performed, no authentication has yet occurred. Therefore, each system has no way of knowing whether or not it is talking to the system it intends. An intruder could attempt to negotiate the use of an authentication system which is either weak, or already compromised by the intruder.

6. Implementation Rules

WILL and DO are used only at the beginning of the connection to obtain and grant permission for future negotiations.

The authentication is only negotiated in one directions; the server must send the "DO", and the client must send the "WILL". This restriction is due to the nature of authentication; there are three possible cases; server authenticates client, client authenticates server, and server and client authenticate each other. By only negotiating the option in one direction, and then determining which of the three cases is being used via the suboption, potential ambiguity is removed. If the server receives a "DO", it must respond with a "WONT". If the client receives a "WILL", it must respond with a "DONT".

Once the two hosts have exchanged a DO and a WILL, the server is free to request authentication information. In the request, a list of supported authentication types is sent. Only the server may send requests ("IAC SB AUTHENTICATION SEND authentication-type-pair-list IAC SE"). Only the client may transmit authentication information via the "IAC SB AUTHENTICATION IS authentication-type ... IAC SE" command. Only the server may send replys ("IAC SB AUTHENTICATION REPLY authentication-type ... IAC SE"). As many IS and REPLY suboptions may be exchanged as are needed for the particular authentication scheme chosen.

If the client does not support any of the authentication types listed in the authentication-type-pair-list, a type of NULL should be used to indicate this in the IS reply. Note that in this case, the server may choose to close the connection.

The order of the authentication types MUST be ordered to indicate a preference for different authentication types, the first type being the most preferred, and the last type the least preferred.

The following is an example of use of the option:

```
    Client                           Server
                                     IAC DO AUTHENTICATION
    IAC WILL AUTHENTICATION
    [ The server is now free to request authentication information.
      ]
                                     IAC SB AUTHENTICATION SEND
                                     KERBEROS_V4 CLIENT|MUTUAL
                                     KERBEROS_V4 CLIENT|ONE_WAY IAC
                                     SE
    [ The server has requested mutual Kerberos authentication, but is
```

```
         willing to do just one-way Kerberos authentication.  The client
         will now respond with the name of the user that it wants to log
         in as, and the Kerberos ticket.  ]
       IAC SB AUTHENTICATION NAME "joe"
       IAC SE
       IAC SB AUTHENTICATION IS
       KERBEROS_V4 CLIENT|MUTUAL AUTH 4
       7 1 67 82 65 89 46 67 7 9 77 0
       48 24 49 244 109 240 50 208 43
       35 25 116 104 44 167 21 201 224
       229 145 20 2 244 213 220 33 134
       148 4 251 249 233 229 152 77 2
       109 130 231 33 146 190 248 1 9
       31 95 94 15 120 224 0 225 76 205
       70 136 245 190 199 147 155 13
       IAC SE
       [ The server responds with an ACCEPT command to state that the
         authentication was successful.  ]
                                        IAC SB AUTHENTICATION REPLY
                                        KERBEROS_V4 CLIENT|MUTUAL ACCEPT
                                        IAC SE
       [ Next, the client sends across a CHALLENGE to verify that it is
         really talking to the right server.  ]
       IAC SB AUTHENTICATION IS
       KERBEROS_V4 CLIENT|MUTUAL
       CHALLENGE xx xx xx xx xx xx xx
       xx IAC SE
       [ Lastly, the server sends across a RESPONSE to prove that it
         really is the right server.
                                        IAC SB AUTHENTICATION REPLY
                                        KERBEROS_V4 CLIENT|MUTUAL
                                        RESPONSE yy yy yy yy yy yy yy yy
                                        IAC SE
```

It is expected that any implementation that supports the Telnet AUTHENTICATION option will support all of this specification.

7. References

[1] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1340, USC/Information Sciences Institute, July 1992.

Security Considerations

Security issues are discussed in Section 5.

Author's Address

David A. Borman, Editor
Cray Research, Inc.
655F Lone Oak Drive
Eagan, MN 55123

Phone: (612) 452-6650
EMail: dab@CRAY.COM

Mailing List: telnet-ietf@CRAY.COM

Chair's Address

The working group can be contacted via the current chair:

Steve Alexander
INTERACTIVE Systems Corporation
1901 North Naper Boulevard
Naperville, IL 60563-8895

Phone: (708) 505-9100 x256
EMail: stevea@isc.com