Telnet Authentication: Kerberos Version 4

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 draft documents valid for a maximum of six
   months. Internet Drafts may be updated, replaced, or obsoleted by
   other documents at any time. It is not appropriate to use Internet
   Drafts as reference material or to cite them other than as a "working
   draft" or "work in progress."

   Please check the 1id-abstracts.txt listing contained in the
   internet-drafts Shadow Directories on nic.ddn.mil, nnsc.nsf.net,
   nic.nordu.net, ftp.nisc.sri.com, or munnari.oz.au to learn the
   current status of any Internet Draft.

1. **Command Names and Codes**

   Authentication Types

       KERBEROS_V4   1

   Suboption Commands

       AUTH   0
       REJECT   1
       ACCEPT   2
       CHALLENGE   3
       RESPONSE   4

2. **Command Meanings**

   IAC SB AUTHENTICATION IS <authentication-type-pair> AUTH <kerberos
   ticket and authenticator> IAC SE

      This is used to pass the Kerberos ticket to the remote side of the

connection. The first octet of the <authentication-type-pair>
value is KERBEROS_V4, to indicate the usage of Kerberos version 4.

IAC SB AUTHENTICATION REPLY <authentication-type-pair> ACCEPT IAC SE

This command indicates that the authentication was successful.

IAC SB AUTHENTICATION REPLY <authentication-type-pair> REJECT <op-
tional reason for rejection> IAC SE

This command indicates that the authentication was not successful,
and if there is any more data in the sub-option, it is an ASCII
text message of the reason for the rejection.

IAC SB AUTHENTICATION IS <authentication-type-pair> CHALLENGE <en-
crypted challenge> IAC SE
IAC SB AUTHENTICATION REPLY <authentication-type-pair> RESPONSE <en-
crypted response> IAC SE

These two commands are used to perform mutual authentication.
They are only used when the AUTH_HOW_MUTUAL bit is set in the
second octet of the authentication-type-pair. After successfully
sending an AUTH and receiving an ACCEPT, a CHALLENGE is sent. The
challenge is a random 8 byte number with the most significant byte
first, and the least significant byte last. When the CHALLENGE
command is sent, the "encrypted challenge" is the 8-byte-challenge
encrypted in the session key. When the CHALLENGE command is re-
ceived, the contents are decrypted to get the original 8-byte-
challenge, this value is then incremented by one, re-encrypted
with the session key, and returned as the "encrypted response" in
the RESPONSE command. The receiver of the RESPONSE command de-
crypts the "encrypted response", and verifies that the resultant
value is the original 8-byte-challenge incremented by one.

The "encrypted challenge" value sent/received in the CHALLENGE
command is also encrypted with the session key on both sides of
the session, to produce a random 8-byte key to be used as the de-
fault key for the ENCRYPTION option.

[3]. Implementation Rules

If the second octet of the authentication-type-pair has the AUTH_WHO
bit set to AUTH_CLIENT_TO_SERVER, then the client sends the initial
AUTH command, and the server responds with either ACCEPT or REJECT.
In addition, if the AUTH_HOW bit is set to AUTH_HOW_MUTUAL, and the
server responds with ACCEPT, then the client then sends a CHALLENGE,
and the server sends a RESPONSE.

If the second octet of the authentication-type-pair has the AUTH_WHO

   bit set to AUTH_SERVER_TO_CLIENT, then the server sends the initial
   AUTH command, and the client responds with either ACCEPT or REJECT.
   In addition, if the AUTH_HOW bit is set to AUTH_HOW_MUTUAL, and the
   client responds with ACCEPT, then the server then sends a CHALLENGE,
   and the client sends a RESPONSE.

   The authenticator (Kerberos Principal) used is of the form
   "rcmd.host@realm".

[4](#). **Examples**

   User "joe" may wish to log in as user "pete" on machine "foo". If
   "pete" has set things up on "foo" to allow "joe" access to his ac-
   count, then the client would send IAC SB AUTHENTICATION NAME "pete"
   IAC SE IAC SB AUTHENTICATION IS KERBEROS_V4 AUTH <joe's kerberos
   ticket> IAC SE The server would then authenticate the user as "joe"
   from the ticket information, and since "pete" is allowing "joe" to
   use his account, the server would send back ACCEPT. If mutual au-
   thentication is being used, the the client would send a CHALLENGE,
   and verify the RESPONSE that the server sends back.

        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 Version 4 Kerberos
          authentication. If mutual authentication is not supported,
          then the server is willing to do one-way 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
        "pete" IAC SE
        IAC SB AUTHENTICATION IS
        KERBEROS_V4 CLIENT|MUTUAL AUTH
        <kerberos ticket information>
        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
```

Author's Address

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

   Phone: (612) 452-6650

   Mailing List: telnet-ietf@CRAY.COM
   EMail: dab@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