

## Telnet Authentication and Encryption 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), `munniari.oz.au` (Pacific Rim), or `nic.nordu.net` (Europe).

### Abstract

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 the clear through the network. This document specifies the AUTH\_ENCRYPT option, whose purpose is two-fold: to provide a framework for the passing of authentication information through the TELNET session and to provide a mechanism to enable encryption of the data stream as a side effect of successful authentication.

### Acknowledgements

This document represents a revision of [RFC1416](#) [1] that has been enhanced to include the optional encryption of the data stream. The work of Dave Borman (Editor), Steve Alexander (Telnet working group chair), and the Telnet working group in the preparation of that document is gratefully acknowledged. Ted Ts'o deserves special mention for keeping track of the details of this enhancement and

INTERNET-DRAFT

Telnet Security Options

July 1995

providing them to us so that this document could be prepared.

## 1. Introduction

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 the clear through the network. If the connection goes through untrusted networks, there is the possibility that an intruder may eavesdrop on the packets as they go by and thus obtain the password.

This document specifies the AUTH\_ENCRYPT option, whose purpose is two-fold: to provide a framework for the passing of authentication information through the TELNET session and to provide a mechanism to enable encryption of the data stream as a side effect of successful authentication. This means that:

- 1) the user's password will not be sent unencrypted across the network,
- 2) if the front end telnet process has the appropriate authentication information it can automatically send it instead of the user typing any password, and
- 3) once authentication has succeeded the data stream can be encrypted to provide protection against active attacks.

It is intended that the AUTH\_ENCRYPT option be general enough that it can be used to pass information for any authentication and encryption type.

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

## 2. Command Names and Codes

This section lists the codes for the Telnet authentication and encryption option, commands, and modifiers, as well as the initially defined authentication and encryption types. The codes for the authentication and encryption types are officially assigned and maintained by IANA. The values are published regularly in STD2, which is currently [RFC1700](#) [2].

AUTH\_ENCRYPT                      XX

#### Authentication and Encryption Commands

IS	0
SEND	1
REPLY	2

J. Galvin, et. al.

Expires January 1996

[Page 2]

---

INTERNET-DRAFT

Telnet Security Options

July 1995

NAME	3
END_ENCRYPT	4
REQUEST_END_ENCRYPT	5

#### Authentication and Encryption Types

NULL	0
KERBEROS_V4	1
KERBEROS_V5	2
SPX	3
RSA	6
LOKI	10
GSSAPI	XX

#### 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

ENCRYPT_MASK	4
ENCRYPT_OFF	0
ENCRYPT_ON	4

INI_CRED_FWD_MASK	8
INI_CRED_FWD_OFF	0
INI_CRED_FWD_ON	8

### [3.](#) 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 AUTH\_ENCRYPT

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

#### IAC DO AUTH\_ENCRYPT

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

encryption information.

#### IAC WONT AUTH\_ENCRYPT

The client side of the connection sends this command to indicate that it refuses to send or receive authentication or encryption information. The server side sends this command if it receives a DO AUTH\_ENCRYPT command and it refuses to send or receive authentication or encryption information.

#### IAC DONT AUTH\_ENCRYPT

The server side of the connection sends this command to indicate that it refuses to send or receive authentication or encryption information. The client side sends this command if it receives a WILL AUTH\_ENCRYPT command and it refuses to send or receive authentication or encryption information.

#### IAC SB AUTH\_ENCRYPT SEND authentication-type-pair-list IAC SE

The server side of the connection sends this command to request that the client side of the connection send authentication information for one of the authentication types listed in the "authentication-type-pair-list". The "authentication-type-pair-list" is an ordered list of "authentication-type" pairs.

IAC SB AUTH\_ENCRYPT IS authentication-type-pair authentication-data  
IAC SE

The client side of the connection sends this command to indicate the "authentication-type-pair" that was chosen for the connection from the list provided by the server side and to send the authentication-data necessary for it.

IAC SB AUTH\_ENCRYPT REPLY authentication-type-pair authentication-data IAC SE

The server side of the connection sends this command in response to authentication-data in a previous IS command and to send the authentication-data the client needs.

IAC SB AUTH\_ENCRYPT NAME remote-user IAC SE

The client side of the connection sends this command to indicate the account name (remote-user) on the server that the user wishes to be authorized to use. Note that while authentication may succeed, the authorization to use a particular account may fail. Some authentication types may ignore this command. This command supercedes the value of the USER environment variable if it is passed from the client to the server.

IAC SB AUTH\_ENCRYPT END\_ENCRYPT IAC SE

The sender of this command is stating that at this point in the data stream, all following data will no longer be encrypted. This command should only be sent in an encrypted data stream and should be ignored if received in an unencrypted data stream. See the "Implementation Rules" section for more details.

IAC SB AUTH\_ENCRYPT REQUEST\_END\_ENCRYPT IAC SE

The sender of this command requests that the remote side stop encryption of the telnet data stream. This command is advisory only. This command should only be sent in an encrypted data stream and should be ignored if received in an unencrypted data stream. See the "Implementation Rules" section for more details.

The "authentication-type-pair" is two octets: the first is the authentication type and the second is a modifier to the type. There are currently four one bit fields defined in the modifier. Two of these are processed as a pair: the AUTH\_WHO\_MASK bit and the AUTH\_HOW\_MASK bit. There are four possible combinations of these two bits:

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 to which it wants to be connected.

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 to which it wants to be connected.

AUTH\_SERVER\_TO\_CLIENT

AUTH\_HOW\_MUTUAL

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

the user on the client side of the connection.

The third bit field in the modifier is the ENCRYPT\_MASK bit. This bit is either set to ENCRYPT\_ON or ENCRYPT\_OFF. Setting this bit to ENCRYPT\_ON implies that once authentication completes, the data stream is to be encrypted in both directions using the encryption method specified for the authentication type.

The fourth bit field in the modifier is the INI\_CRED\_FWD\_MASK bit. This bit is either set to INI\_CRED\_FWD\_ON or INI\_CRED\_FWD\_OFF. Setting this bit to INI\_CRED\_FWD\_ON implies that once authentication completes, the client will immediately forward authentication credentials to the server. This bit is set by the client to advise the server to expect forwarded credentials from the client.

The motivation for this advisory bit is that the server may wish to wait until the forwarded credentials have been sent before starting any operating system specific login procedures which may depend on these credentials. Note that credentials forwarding may not be supported by all authentication types. It is a protocol error to set this bit if the underlying authentication type does not support credentials forwarding.

The authentication-data may be omitted if there is none to be provided for the type being negotiated.

#### [4.](#) Default Specification

The default specification for this option is

```
WONT AUTH_ENCRYPT
DONT AUTH_ENCRYPT
```

meaning there will not be any exchange of authentication or encryption information.

#### [5.](#) 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 direction; 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 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 AUTH\_ENCRYPT SEND authentication-type-pair-list IAC SE"). Only the client MAY transmit authentication information via the "IAC SB AUTH\_ENCRYPT IS authentication-type ... IAC SE" command. Only the server MAY send replies ("IAC SB AUTH\_ENCRYPT REPLY authentication-type ... IAC SE"). As many IS and REPLY suboptions MAY be exchanged as are needed for the particular authentication type chosen.

When determining a match from the authentication-type-pair-list received from the server, the client MAY ignore the AUTH\_ENCRYPT\_MASK bit. If the AUTH\_ENCRYPT\_MASK bit was ENCRYPT\_OFF, then the client MUST respond with ENCRYPT\_OFF. If the AUTH\_ENCRYPT\_MASK bit was on, then the client MAY respond with either ENCRYPT\_ON or ENCRYPT\_OFF. In the latter case the client is stating that it will do authentication but it does not want to encrypt the data stream.

If the client does not support any of the authentication types listed in the authentication-type-pair-list, it SHOULD indicate this in the IS reply with a type of NULL. Note, if the client turns off the ENCRYPT\_ON bit or responds with a type of NULL, the server MAY choose to close the connection.

Encryption from the server to the client begins with the first byte immediately following the "IAC SB AUTH\_ENCRYPT REPLY ... IAC SE" command that signifies that the server has successfully completed the authentication process. Encryption from the client to the server begins with the first byte immediately following the "IAC SB AUTH\_ENCRYPT RESPONSE ... IAC SE" command that signifies that the client has successfully completed the authentication process. Both of these will be specified in the document for the specific authentication and encryption type. All data, including TELNET options, are encrypted.

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 being the least preferred.

Special consideration applies to the use of END\_ENCRYPT and REQUEST\_END\_ENCRYPT. A scenario during which one may want to turn off encryption is communication from the server to the client, which has the bulk of the data; leaving the communication from the client to the server encrypted ensures that typed passwords are not readable by eavesdropping. To do this the client SHOULD send a REQUEST\_END\_ENCRYPT command to the server, who SHOULD then send an END\_ENCRYPT command and stop encrypting the output data stream. At this point, an active attacker could insert a REQUEST\_END\_ENCRYPT command in the data stream from the server to the client to try and get the client to stop encrypting its input stream to the server. So, a REQUEST\_END\_ENCRYPT command SHOULD always be honored if received within an encrypted data stream but SHOULD be ignored if received over an unencrypted data stream. If it is desirable to disable all encryption, a REQUEST\_END\_ENCRYPT SHOULD be sent prior to the END\_ENCRYPT to get the other side to stop encrypting first.

## 6. User Interface Rules

Normally protocol specifications do not address user interface issues. However, due to the fact that the user should be able to indicate the information necessary to achieve a successful authentication and encryption negotiation and the user should know whether the authentication and encryption succeeded, some guidance must be given to implementors to assure a minimum level of user control.

The user MUST be able to specify whether or not authentication is to be used and whether or not encryption is to be used if the authentication succeeds. There SHOULD be at least four settings: REQUIRE, PROMPT, WARN, and DISABLE.

Setting the authentication switch to REQUIRE means that if the authentication fails, then an appropriate error message MUST be displayed and the TELNET connection MUST be terminated.

Setting the authentication switch to PROMPT means that if the authentication fails, then an appropriate error message MUST be displayed and the user MUST be prompted for confirmation before continuing the TELNET session.

Setting the authentication switch to WARN means that if the authentication fails, then an appropriate error message MUST be displayed before continuing the TELNET session.

Setting the authentication switch to DISABLE means that authentication MUST NOT be attempted.

The encryption switch SHOULD have an independent set of the same settings as the authentication switch. However, its settings MUST

only be used when authentication succeeds.

The default setting for both switches SHOULD be WARN. Both of these switches MAY be implemented as a single switch, though having them separate gives more control to the user.

## [7.](#) Example

The following is an example of the use of this option for authentication without encryption for Kerberos Version 4 [\[3\]](#):

Client

Server

IAC DO AUTH\_ENCRYPT

IAC WILL AUTH\_ENCRYPT

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

IAC SB AUTH\_ENCRYPT SEND  
KERBEROS\_V4  
AUTH\_CLIENT\_TO\_SERVER|AUTH\_HOW\_MUTUAL  
KERBEROS\_V4  
AUTH\_CLIENT\_TO\_SERVER|AUTH\_HOW\_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 AUTH_ENCRYPT NAME "joe"
IAC SE
IAC SB AUTH_ENCRYPT IS
KERBEROS_V4
AUTH_CLIENT_TO_SERVER|AUTH_HOW_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 AUTH\_ENCRYPT REPLY

J. Galvin, et. al.

Expires January 1996

[Page 9]

---

INTERNET-DRAFT

Telnet Security Options

July 1995

```
KERBEROS_V4
AUTH_CLIENT_TO_SERVER|AUTH_HOW_MUTUAL
ACCEPT IAC SE
```

[ Next, the client sends across a CHALLENGE to verify that it is really talking to the right server. ]

```
IAC SB AUTH_ENCRYPT IS
KERBEROS_V4
AUTH_CLIENT_TO_SERVER|AUTH_HOW_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 AUTH_ENCRYPT REPLY
KERBEROS_V4
AUTH_CLIENT_TO_SERVER|AUTH_HOW_MUTUAL
RESPONSE yy yy yy yy yy yy yy yy
IAC SE
```

The following is an example of the use of this option for authentication with encryption for Kerberos Version 4 [3]:

Client

Server

IAC WILL AUTH\_ENCRYPT

IAC DO AUTH\_ENCRYPT

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

IAC SB AUTH\_ENCRYPT SEND  
KERBEROS\_V4  
AUTH\_CLIENT\_TO\_SERVER|AUTH\_HOW\_MUTUAL  
|ENCRYPT\_ON KERBEROS\_V4  
AUTH\_CLIENT\_TO\_SERVER|AUTH\_HOW\_ONE\_WAY  
|ENCRYPT\_ON IAC SE

[ The server has requested mutual Kerberos authentication but is willing to do just one-way Kerberos authentication. In both cases it is willing to encrypt the data stream. The client will now respond with the name of the user that it wants to log in as and the Kerberos ticket. ]

IAC SB AUTH\_ENCRYPT NAME "joe"  
IAC SE  
IAC SB AUTH\_ENCRYPT IS  
KERBEROS\_V4

J. Galvin, et. al.

Expires January 1996

[Page 10]

---

INTERNET-DRAFT

Telnet Security Options

July 1995

AUTH\_CLIENT\_TO\_SERVER|AUTH\_HOW\_MUTUAL|ENCRYPT\_ON  
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 AUTH_ENCRYPT REPLY
KERBEROS_V4
AUTH_CLIENT_TO_SERVER|AUTH_HOW_MUTUAL
|ENCRYPT_ON ACCEPT IAC SE
```

[ Next, the client sends across a CHALLENGE to verify that it is really talking to the right server. ]

```
IAC SB AUTH_ENCRYPT IS
KERBEROS_V4
AUTH_CLIENT_TO_SERVER|AUTH_HOW_MUTUAL|ENCRYPT_ON
CHALLENGE xx xx xx xx xx xx xx
xx IAC SE
```

[ At this point, the client begins to encrypt the outgoing data stream, and the server, after receiving this command, begins to decrypt the incoming data stream. Lastly, the server sends across a RESPONSE to prove that it really is the right server. ]

```
IAC SB AUTH_ENCRYPT REPLY
KERBEROS_V4
AUTH_CLIENT_TO_SERVER|AUTH_HOW_MUTUAL
|ENCRYPT_ON RESPONSE yy yy yy yy
yy yy yy yy IAC SE
```

[ At this point, the server begins to encrypt its outgoing data stream, and the client, after receiving this command, begins to decrypt its incoming data stream. ]

## [8.](#) Security Considerations

The ability to negotiate a common authentication type between a client and a server system is a feature of the authentication option that should be used with caution. When the negotiation is performed no authentication has yet occurred. Therefore, neither system knows

whether it is communicating with the intended system. An active attacker could attempt to negotiate the use of an authentication system which is either weak or already compromised by the intruder.

By linking the enabling of encryption as a side effect of successful authentication, protection is provided against an active attacker. An active attack is one where the underlying TCP stream can be modified or taken over by an active attacker. If encryption were enabled as a separate negotiation, it would provide a window of vulnerability from when the authentication completes up to and including the negotiation to turn on encryption. It is because of this there is no command to restart encryption. The only safe way to restart encryption once it has been turned off is to repeat the entire authentication process.

## 9. References

- [1] D. Borman, Editor. Telnet Authentication Option. [RFC1416](#), Cray Research, Inc., February 1993.
- [2] J. Reynolds, J. Postel. Assigned Numbers. [RFC1700](#), ISI, October 1994.
- [3] D. Borman, Editor. Telnet Authentication: Kerberos Version 4. [RFC1411](#), Cray Research, Inc., January 1993.

## 10. 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