Telnet Forwarding of X Windows Session Data

0. Abstract

This Internet-Draft describes a mechanism via which X Windows client applications to which a telnet session has been established may have their communications with the X Windows Server forwarded via the Telnet communications channel. This is desirable when the Telnet session is established through a Firewall or Network Address Translator which does not allow arbitrary connections to be created from the host machine to the client machine; or when the Telnet session is using an authenticated and encrypted channel and that same security is desired for the X Windows session data.

1. Command Names and Codes

FORWARD_X 49 (assigned by IANA)

Sub-option Commands

FWDX_SCREEN 0
FWDX_OPEN 1
FWDX_CLOSE 2
FWDX_DATA 3

2. Command Meanings

IAC WILL FORWARD_X

The server side of the connection sends this command to indicate that it is willing to send and receive X Windows session data via the telnet connection. The client must not send this command.

IAC DO FORWARD_X
The client side of the connection sends this command to indicate that it is willing to send and receive X Windows session data via the telnet connection. The server must not send this command.

IAC WONT FORWARD_X

The server side of the connection sends this command to indicate that it is not willing or able to send and receive X Windows session data via the telnet connection. If the client receives IAC DO FORWARD_X it must respond with IAC WONT FORWARD_X.

IAC DONT FORWARD_X

The client side of the connection sends this command to indicate that it is not willing or able to send and receive X Windows session data via the telnet connection. If the server receives IAC WILL FORWARD_X it must respond with IAC DONT FORWARD_X.

IAC SB FORWARD_X FWDX_SCREEN <screen> IAC SE

The client side of the connection sends this command to the server to indicate to the server the screen (or monitor) number being used by the local X Windows server. <screen> is a single octet with legal values of 0 to 255. The screen number is to be used by the server when constructing the DISPLAY environment variable to be used on the host.

The server side of the connection must not send this command.

IAC SB FORWARD_X FWDX_OPEN <channel> IAC SE

The server side of the connection sends this command to the client to indicate that a new X Windows session is being started and that a new channel should be allocated. <channel> is two octets in network byte order.

The client side of the connection must not send this command.

IAC SB FORWARD_X FWDX_CLOSE <channel> IAC SE

Either side of the connection sends this command to indicate to the other that the channel has been terminated and that the associated resources should be freed. <channel> is two octets in network byte order.

IAC SB FORWARD_X FWDX_DATA <channel> <data> IAC SE

Either side of the connections sends this command to the other to forward X Windows session data across the Telnet connection. <channel> is two octets in network byte order. <data> is an arbitrary length stream of bytes. All occurrences of 0xFF in the data stream must be doubled to avoid confusion with telnet commands.

3. Default Specification

The default specification for this option is

WONT FORWARD_X
DONT FORWARD_X

meaning there will not be any forwarding of X Windows session data.
4. Motivation

Firewalls and Network Address Translators sometimes make it impossible for X Windows clients to connect to the local X Windows server. In these situations it is necessary to have a method to forward (or tunnel) the data along a connection which is already established.

When Telnet Authentication and Encryption or Telnet over TLS are in use it is desirable to afford the same level of protection to the X Windows session data that is afforded to the Telnet session data.

This option provides a mechanism for using the Telnet connection as a tunnel which then applies its own level of security to the X Windows sessions.

5. Implementation Rules

WILL and DO are negotiated only at the beginning of the Telnet session to obtain and grant permission for future FORWARD_X sub-negotiations. After WILL and DO are exchanged the client must send a FWDX_SCREEN negotiation so the server may establish the appropriate DISPLAY environment variable.

After receipt of FWDX_SCREEN the server will define a DISPLAY variable on the host which shall cause all future X Windows sessions created within that Telnet session to be redirected to the Telnet server. This DISPLAY variable must point to a socket or other mechanism via which the Telnet Server will be able to listen for new X Windows sessions.

Whenever the server accepts a new X Windows session it allocates a new channel and sends a FWDX_OPEN negotiation to the client. The client allocates any necessary resources for the support of the channel and opens a local connection to the X Windows Server specified by the local environment.

All data read by the server from the X Windows clients or from the X Windows Server by the client are forwarded to the peer via the use of a FWDX_DATA negotiation.

When the X Windows client closes the connection the server will send a FWDX_CLOSE negotiation to the client. If the X Windows Server closes the connection the client with send a FWDX_CLOSE to the server.

The Telnet server should not allocate X Windows display number 0 but instead should leave it available for the local X Windows server on the same machine.

The Telnet client should not negotiation FORWARD_X if it does not have a local X Windows server available.

FORWARD_X takes precedence over Telnet X-Display Location and the DISPLAY variable transmitted via Telnet Environment. If FORWARD_X has been negotiated prior to the receipt of other display information, this subsequent information must be ignored.

6. Security Considerations

FORWARD_X is independent of Telnet Authentication and Encryption, and Telnet over TLS. Use of FORWARD_X without the use of Telnet Authentication and Encryption or Telnet over TLS does not provide any privacy benefits.

When the Telnet Server creates a socket to listen for new X Windows clients it should ensure that the connections it accepts have originated on the same machine on which it is executing (and when possible verify the
identity of the user making the connection.) Otherwise, the listen socket may be used to gain access via an otherwise secure channel to the Telnet client’s X Windows server.

7. Example

Initial negotiations

S: IAC WILL FORWARD_X
C: IAC DO FORWARD_X IAC SB FORWARD_X FWDX_SCREEN 00 IAC SE

Server established a listen socket on port 6001 (display 1) and puts an DISPLAY=<ip-address>:<display>:<screen> (i.e. 127.0.0.1:1:0) variable into the local environment.

The server receives a connection from an X Windows client and allocates channel 0:

S: IAC SB FORWARD_X FWDX_OPEN 00 00 IAC SE

Client creates connection to local X Windows server.

Server receives data to send from X Windows client to X Windows server.

S: IAC SB FORWARD_X FWDX_DATA 00 00 <data> IAC SE

X Windows server replies:

C: IAC SB FORWARD_X FWDX_DATA 00 00 <data> IAC SE

X Windows client closes the connection:

S: IAC SB FORWARD_X FWDX_CLOSE 00 00 IAC SE

Authors’ Addresses

Jeffrey Altman
Columbia University
Watson Hall Room 716
612 West 115th Street
New York NY 10025
Phone: +1 (212) 854-1344
EMail: jaltman@columbia.edu

Peter Runestig
Orsangesvagen 83
821 50 Bollnas
Sweden
Phone: +46-278-35777
EMail: peter@runestig.com

Mailing List: telnet-wg@bsdi.com

Jeffrey Altman * Sr.Software Designer * Kermit-95 for Win32 and OS/2
The Kermit Project * Columbia University
612 West 115th St #716 * New York, NY * 10025
http://www.kermit-project.org/k95.html * kermit-support@kermit-project.org