CprE 530

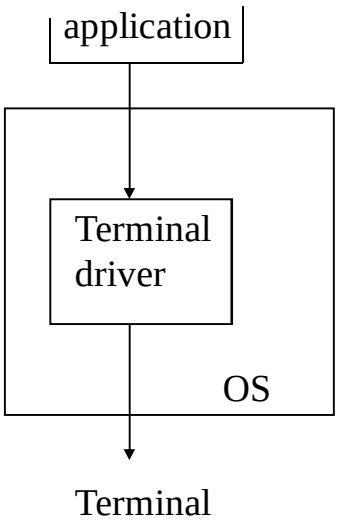
Lecture 22

Topics

- Remote Access
- Telnet
  - Network Virtual Terminal (NVT)
  - Commands
  - Option Negotiation
- Rlogin
- X-Windows

Telnet

TELNET: a Virtual Terminal Protocol that provides interactive access to remote computers

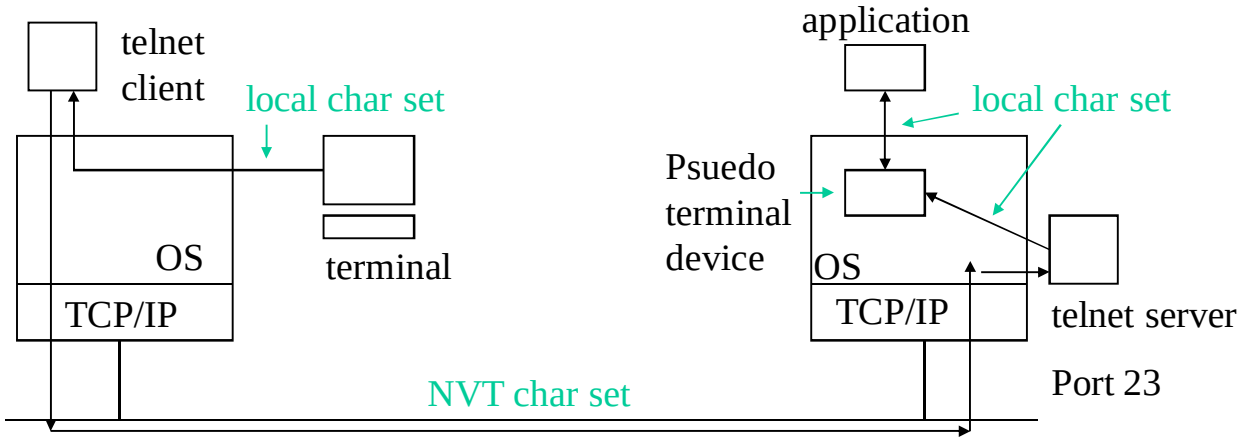


The protocol defines:

- Format of data
- How control signals are passed and how to distinguish them from data
- Data transfer mode
  (half/full duplex, sync/async)
- How out-of-band signals are passed
- How data delivery is controlled

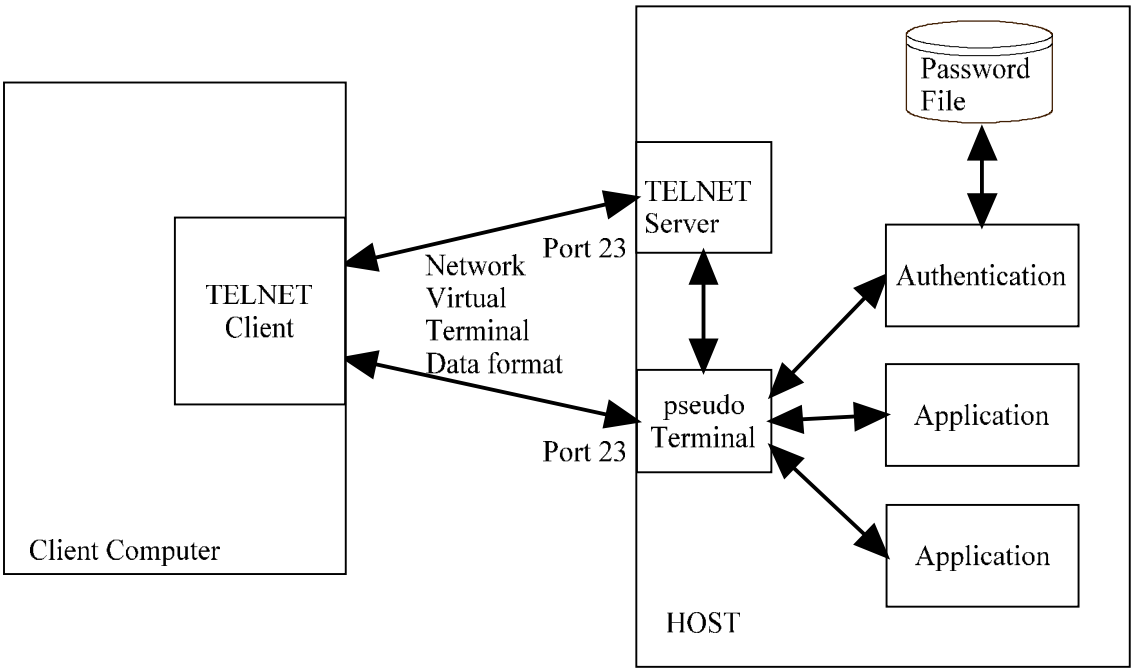
Telnet

NVT – Network Virtual Terminal



Local charsets of different OS's may not be compatible. When sending over the network, the local charset is translated to the common NVT charset by the telnet client. The telnet server then translates the NVT charset to the local charset

Telnet



Telnet

The virtual terminal consists of a display and a printer

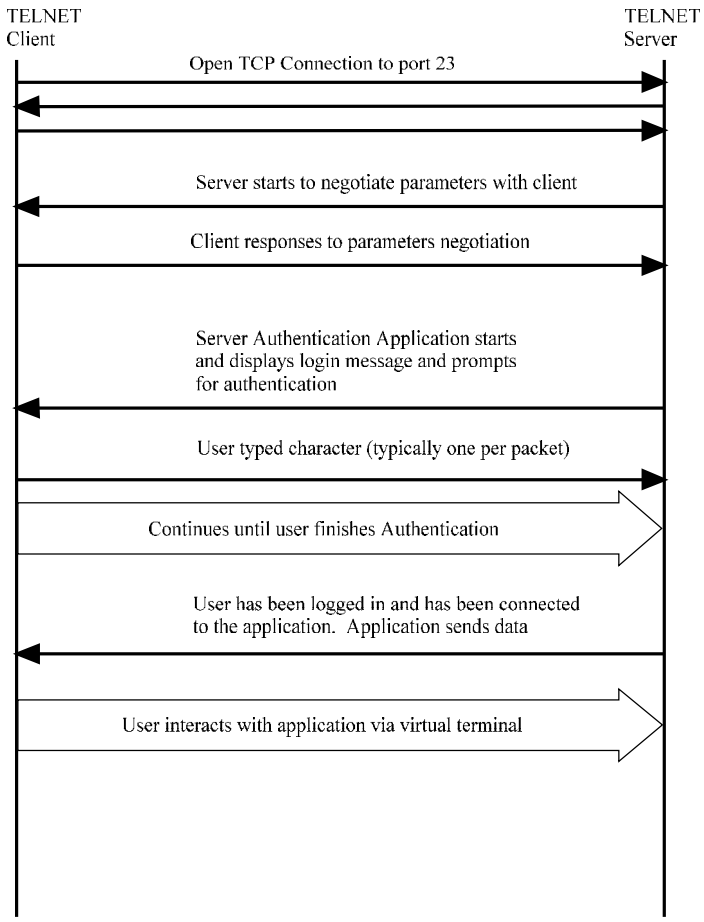
- Display
  - Characters are 7 bit ASCII
  - Operates in scroll mode with unlimited line length, unlimited lines per page
  - Must be able to generate control signals:

    Are You There Interrupt Process

    Abort Output Erase Character

    Erase line Break
- Printer
  - Has unspecified line width and page length
  - Can print the 95 ASCII graphic characters
  - Can respond to the control codes:

    NUL Line Feed Carriage return



Telnet

Telnet Commands

Definition	Abbr	code	
End of subnegotiation	SE	240	
No Operation	NOP	241	
Data Mark: A stream sync character	DM	242	
Break	BRK	243	
Interrupt Process		IOP	244
Abort Output	AO	245	
Are You There	AYT	246	
Erase Character		EC	247
Go Ahead: turn line around for half duplex		GA	249
Begin subnegotiation	SB	250	
WILL		251	
WONT		252	
DO		253	
DON'T		254	
Interpret as CMD		IAC	255

Telnet Commands

How to mix user data and commands:

user data:

command:

There is a special command to transfer 8 byte data

Telnet Options

- Options can be negotiated by telnet processes
- New options can be accommodated since they are not part of the standard
- Three categories
  1. Enhance, change, and refine NVT characteristics
     (e.g. line width)
  2. Change transfer protocol
     (e.g. suppress GO AHEAD)
  3. Information to be passed to the host
     (e.g. status, terminal type)

Telnet Options

This is just a subset of the options defined in many different RFC's:

ID	Name	RFC	Category	
0	Binary transmission		856	2
1	echo	857	1	
5	status	859	3	
8	output line width		1	
9	Output page size		1	
10	Output <cr> disposition	652	1	
24	terminal type	930	3	
25	End of record	885	3	

Telnet Negotiation

Option negotiation rules:

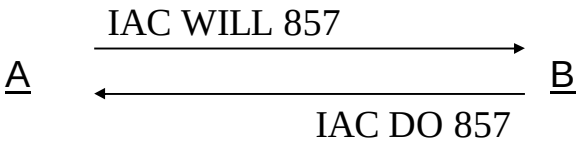
- May reject a request to enable an option
- Must accept a request to disable an option
- Options are not enabled until negotiation is complete
- Never negotiate an option that is already true

Telnet Negotiation

Option negotiation commands:

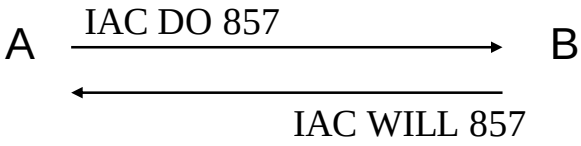
- WILL — Sender wants to enable the option
- WONT — Sender does not want to enable the option
- DO option — Sender would like the other side to enable the
- DON'T option — Sender would not like the other side to enable the

Example 1: Side A wants to enable ECHO (857), side B agrees

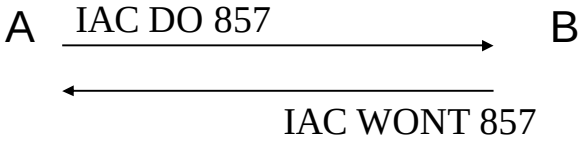


Telnet Negotiation

- Example 2: A would like B to enable ECHO, B agrees

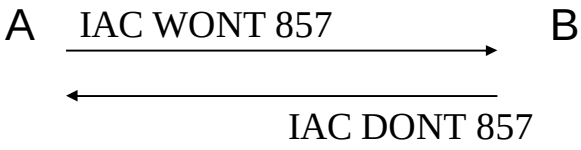


- Example 3: A would like B to enable ECHO, but B does not agree

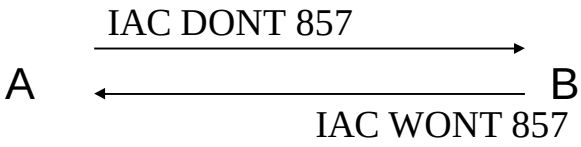


Telnet Negotiation

- Example 4: A would like to disable echo, B MUST agree



- Example 5: A would like B to disable echo, B must agree

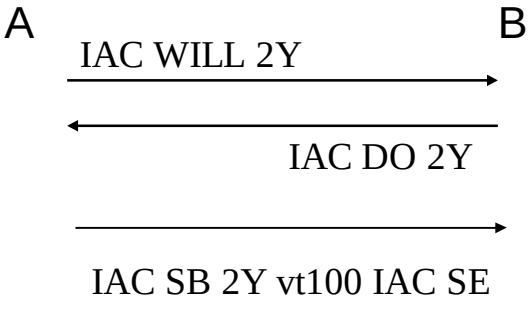


Telnet Negotiation

Suboptions

SE	240	suboption end
SB	241	suboption begin

Example: A wants to set the terminal type (2Y) to vt100

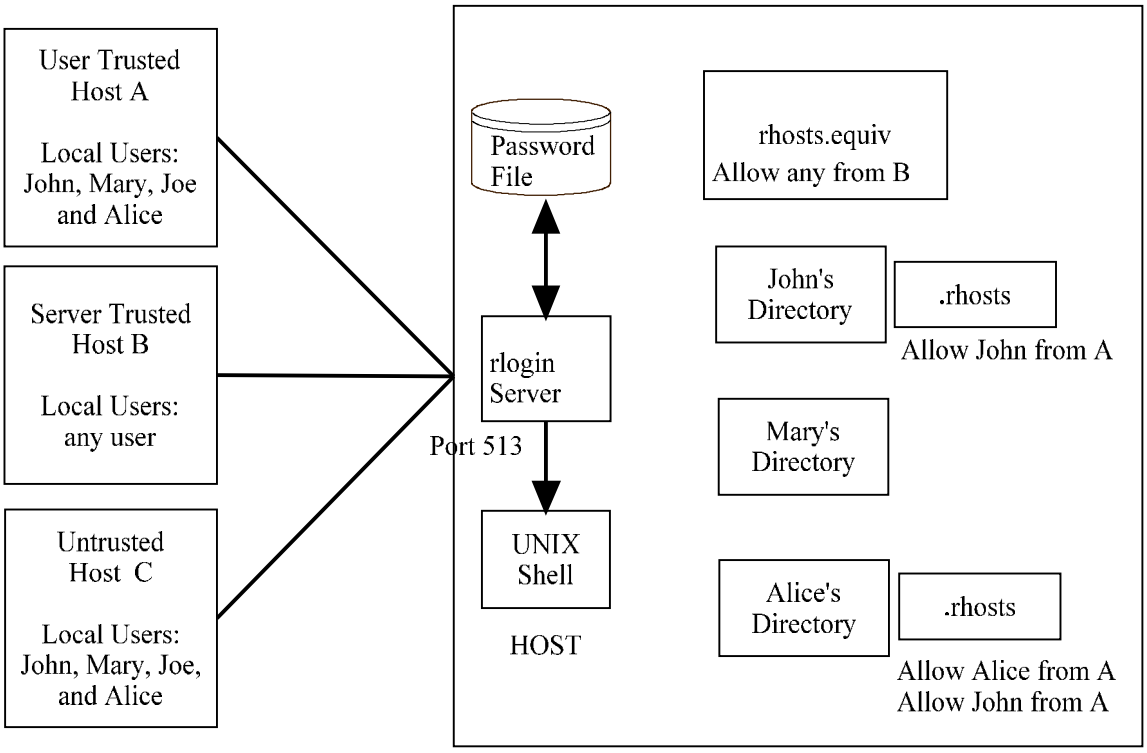


Direction	Data	Comments
C ß S	0xff 0xfd 0x01 0xff 0xfd 0x22 0xff 0xfb 0x05	IAC, Do Echo (request client echoes) IAC, Do linemode (request client sends a line at a time) IAC, Will Status (server wishes to send status info)
C à S	0xff 0xfb 0x01 0xff 0xfc 0x22 0xff 0xfe 0x05	IAC, Will Echo (client will echo characters) IAC, Won't linemode (Client will not do linemode) IAC, Don't Status (client does not want server to send status information)
C ß S	0xff 0xfe 0x01 0xff 0xfb 0x01	IAC, Don't Echo (tell client not to echo) IAC, Will Echo (tell client server will echo)
C à S	0xff 0xfc 0x01 0xff 0xfd 0x01	IAC, Won't Echo (tell server client will not echo) IAC, Do Echo (tell server it is OK to echo)
C ß S	\r\n Login:	Send authentication application prompt
C à S	j	First char of user name
C ß S	j	Echo of the character
		Repeat until enter key is pressed
C à S	\r\n	Send carriage return + linefeed
C ß S	\r\n	Echo carriage return + linefeed
C ß S	Password:	Send authentication application prompt
C à S	p	First char of password (server will not echo)
		Repeat until enter key is pressed
C à S	\r\n	Send carriage return + linefeed
C ß S	\r\n	Echo carriage return + linefeed
C ß S		User is now connected and server application will send message.

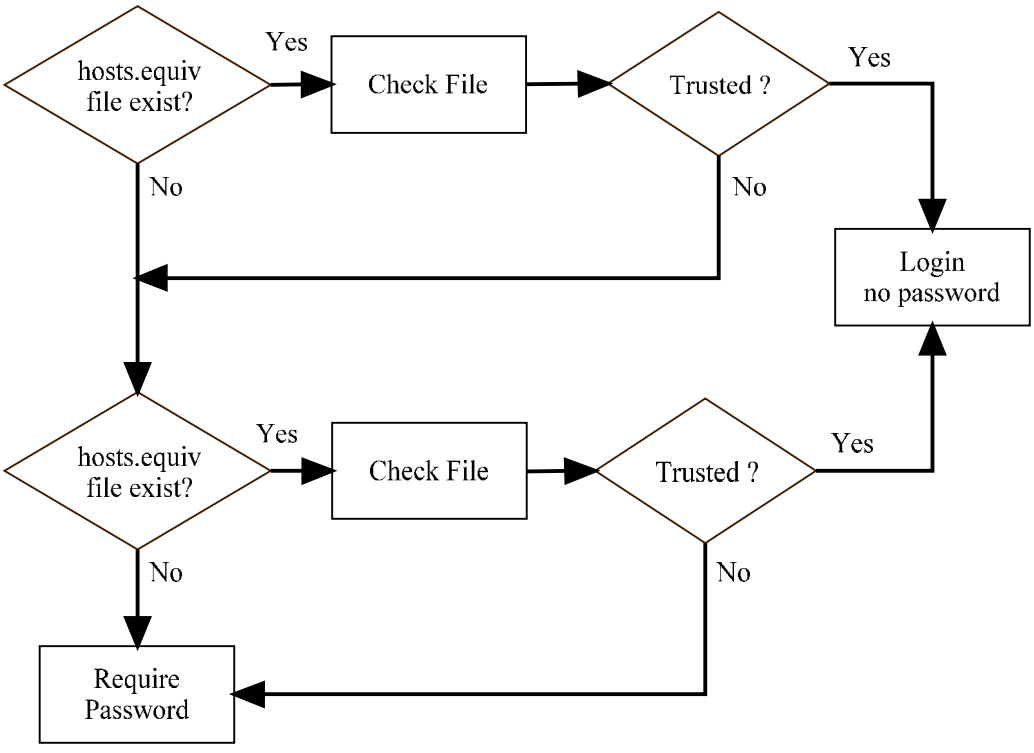
Rlogin

- Remote login (rlogin)
- Similar to telnet, but much simpler
- Designed for unix to unix communication
- Possible for hosts to login without a password
- Uses port 513
- Sequence:
  - Client sends: \0
local login name
\0
server login name
\0
terminal type
\0
  - Server sends: \0

Rlogin



rlogin server trust



rlogin trust

Client host	Client side user	Server side user	Result
A	John	John	Trusted
		Mary	Not Trusted
		Alice	Trusted
	Mary	John	Not Trusted
		Mary	Not Trusted
		Alice	Not Trusted
	Joe	John	Not Trusted
		Mary	Not Trusted
		Alice	Not Trusted
	Alice	John	Not Trusted
		Mary	Not Trusted
		Alice	Trusted
B	Any User	Any User	Trusted

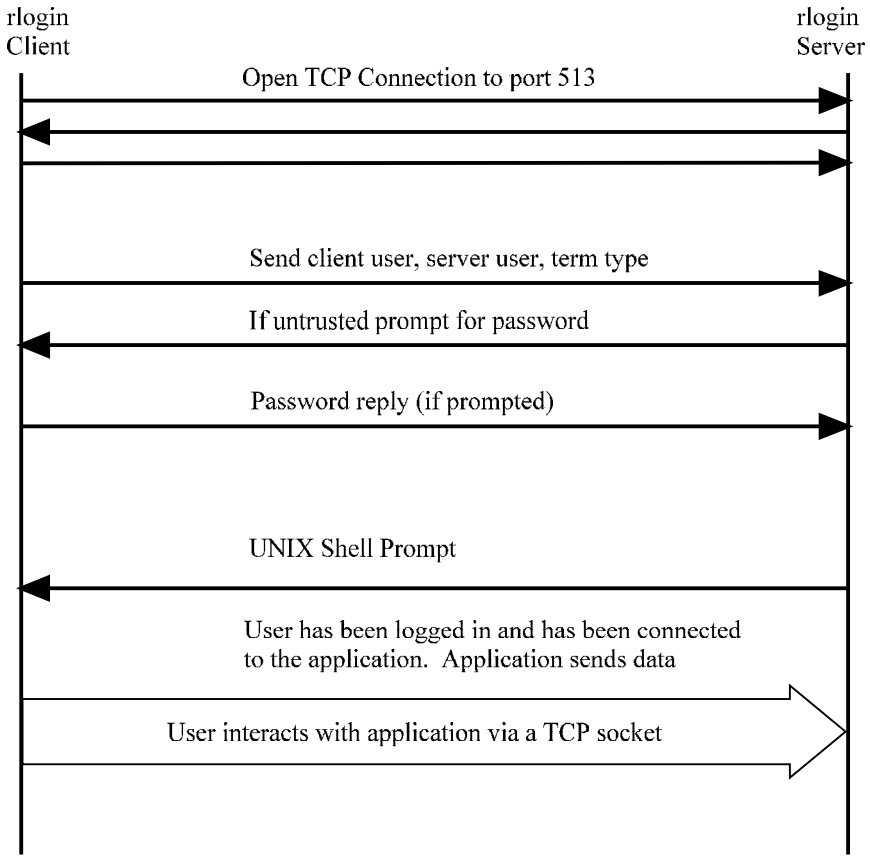
rlogin trust

Client host	Client side user	Server side user	Result
C	John	John	Not Trusted
		Mary	Not Trusted
		Alice	Not Trusted
	Mary	John	Not Trusted
		Mary	Not Trusted
		Alice	Not Trusted
	Joe	John	Not Trusted
		Mary	Not Trusted
		Alice	Not Trusted
	Alice	John	Not Trusted
		Mary	Not Trusted
		Alice	Not Trusted

Rlogin commands

- Commands are distinguished by 0xFF
  - Remote flow control 0x10
  - Local flow control 0x20
  - Window size 0x80

  (asks client for current window size)
- Escape character: ~ ^d
- Everything is sent in clear text



rlogin

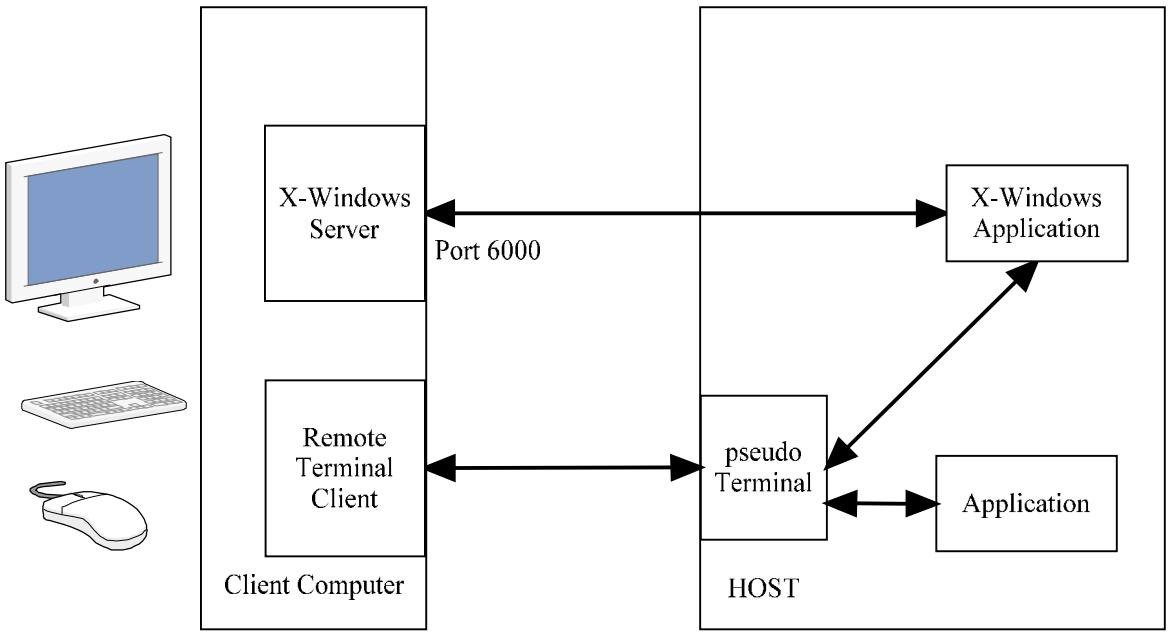
rlogin

Direction	Data	Comments
C **à** S	john 0x00 john 0x00 xterm\34800 0x00	Client side username Server side username Terminal type and speed
		If authentication is required (user is untrusted)
C **ß** S	Password:	Prompt for password
C **à** S	p	First char of password (server will not echo)
		Repeat until enter key is pressed
C **à** S	\r	Send carriage return
C **ß** S	\r\n	Echo carriage return + linefeed
		If authentication worked or user was trusted
C **ß** S	Data from server	User is now connected and server will display the UNIX shell prompt.

X windows

- The user sits on the server side of X windows
  - Usually telnet into client and start X window client
  - X windows then starts and the client authenticates to the X windows server
  - X windows sends information in clear text

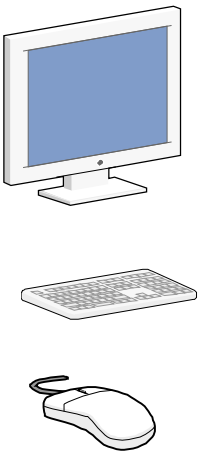
X-windows

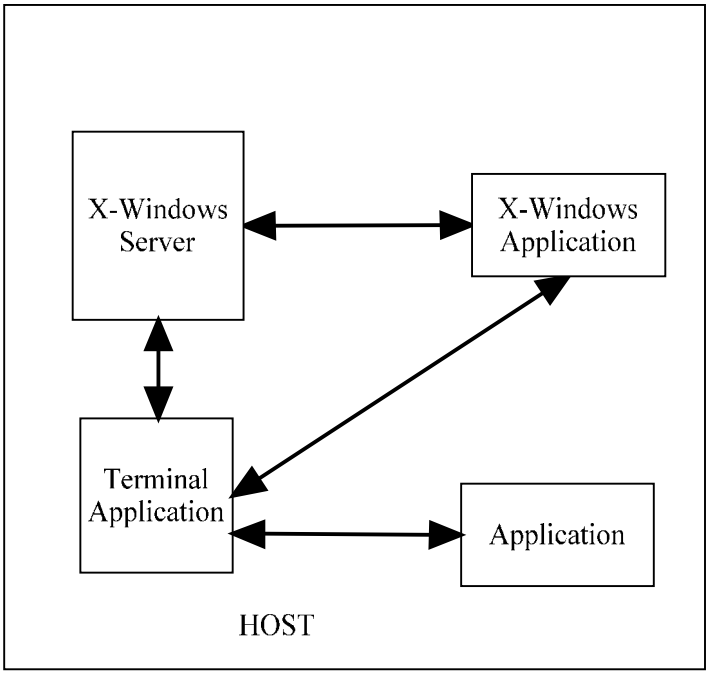


Communication

- In order for two programs to communicate in Unix, a pipe is created between the two processes
  - Pipe works like it sounds, put data in on one side comes out the other
  - Pipe created in the tmp directory
- Port 6000

Local X-Windows

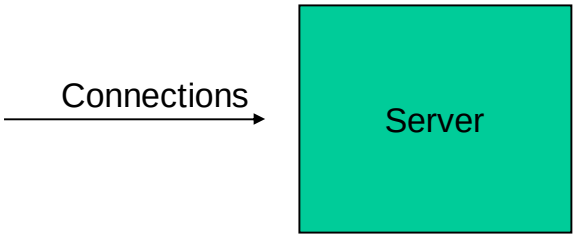




Server Side

- X windows offers up your computer to the outside world to manipulate
- Pc also has public domain X windows programs
- Xhost determines who can connect to your server
  - Xhost + would allow all to connect to one's X windows
- X windows is designed to allow applications control over the display
- Client side
  - How does client know which server to connect to
  - Variable called display
    - :0.0 display means local display
    - The second number is the monitor
    - If remote machine:0.0 which is set on the client
      - Tells X windows to point to server

Server Side cont…



- Authentication?
  - Xhost command, indicates who can connect to one's server, which is IP address based authentication
  - Xhost + allow all connections
  - Xhost - allows nobody
- Command set is designed to allow total control over input and display
  - Through X windows, hackers could
    - Capture screen
    - Capture keystrokes
    - Create, destroy windows
    - Enter key strokes into windows

Local Side

- Pipe
  - /tmp/.X11 …
  - Tmp directory is shared and is world read writable
  - Can do denial of service by deleting the pipe in the tmp directory
    - No new clients can connect
    - Current clients stay connected

Header Based

- For Telnet and rlogin there is not much of a header.
- X-Windows there is possible buffer overflow attacks.

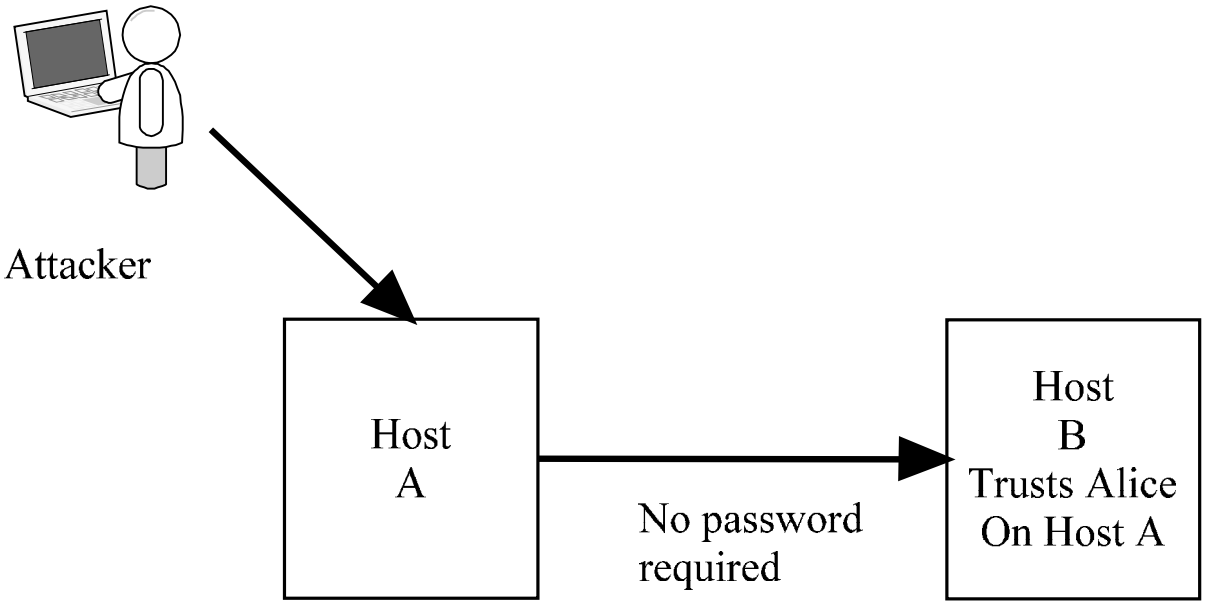
Protocol Based

- Telnet and rlogin have a simple protocol and there is not any attacks, other than telnet can be used to connect to any service (not really a flaw)
- X-Windows has some issues with the protocol since the protocol gives the application control over the remote computer.

Authentication Based

- Telnet offers access to the remote machine and to the login prompt
- Rlogin does not need password unless setup correctly. Uses IP address for authenticator
- X-Windows
  - server can allow any machine to control it based on the IP address
  - Client uses machine authentication to allow a user to run the application

Authentication Stepping stone



Traffic Based

- All three are clear text (sniffing)
  - Usernames & Password
  - Commands and text