Agenda

- A brief introduction to TELNET
- Concept of remote/virtual terminal
- TELNET operations
- TELNET protocol
- TELNET options negotiation
- Other remote access technologies
A Brief Introduction to TELNET
Brief Introduction To TELNET

- Use of TELNET: BBS
- What is TELNET?
- History
- Major Ideas
BBS: Bulletin Board System

- A computer system running software that allows users to connect and log in to the system using a terminal program.
- Once logged in, a user can perform functions such as
  - uploading and downloading software and data;
  - reading news and bulletins;
  - and exchanging messages with other users, either through Email or in public message boards.
Traditional BBS example:

Scanning makes it easy to find files that have been added to since you last read them. Simply press [SPACE] in a menu to find the next updated file. For more information, press [?] [C] [M] from this screen.

Welcome to the new version of Monochrome!
(version 1.101j)

Menu [ESC] = Utilities (inc. Talker & EXIT)
Menu [I] = Help and Information on Monochrome

Menu [N] = News and Media
Menu [T] = Science, Technology and Medicine
Menu [E] = Entertainment
Menu [C] = Society and Culture
Menu [R] = Recreation

Menu [M] = Monochrome Users

Hello 'Guest User'. (guest2:2)
<< 5 other users at Tue Jun 22 03:36 GMT (You have new messages) >>
Example of BBS Today(1)
Example of BBS Today(2)
telnet BBS of BYR
What Is TELNET? (1)

- A protocol used to establish a **dumb terminal** session to another computer on the Internet
- An important Internet **application** for remote access
What Is TELNET? (2)

- Definition in RFC854
  - The purpose of the TELNET Protocol is to provide a general, bi-directional, byte oriented communications facility.
  - Its primary goal is to allow a standard method of interfacing terminal devices and terminal-oriented processes to each other.
  - It is envisioned that the protocol may also be used for terminal-terminal communication ("linking") and process-process communication (distributed computation).
TELNET vs. telnet

- **TELNET** is a *protocol* that provides “a general, bi-directional, eight-bit byte oriented communications facility”
- **telnet** is a *program* that supports the TELNET protocol over TCP
- Many application protocols are built upon the TELNET protocol
The History Of Telnet

- Telnet is simple
  - Total pages of RFC 854 is 15
  - HTTP (we will see later) is 176 pages

- The idea of **option negotiation** was a very good design feature
  - Enables telnet to evolve to meet new demands without endless new versions of basic protocol

- Currently over 100 RFCs on telnet and its options
Major Ideas Of TELNET

- The concept of a **NVT (Network Virtual Terminal)**
  - Providing a standard interface to remote systems
- The principle of **negotiated options**
  - Enabling Telnet to evolve to meet new demands without endless new versions of basic protocol
- A **symmetric view** of terminals and processes
  - Allowing an arbitrary program to become a client
Network Virtual Terminal

Standard language

Connection

Command/Data
Concept Of Remote / Virtual Terminal
Remote Terminal Access

- Early motivation for networks was remote access to interactive systems

- **Dumb terminals** (see figure on the next slide)
  - Keyboard and screen with primitive communication hardware
  - Local host computer establish connection to remote host

- The challenge is that terminals and host systems were not standardized
  - Local terminal was not speaking the same language as the remote host
Telnet Operation Environment On Early Internet
Problem

- Lack of common language between the terminal and the remote host

Client System format

Server System format

TCP connection across Internet
Network Virtual Terminal

- The approach to solve the problem of lack of a common language was to define a common language
  - Virtual terminal protocol (VTP)
- Transform local characteristics into standardized form
  - Network virtual terminal (NVT)
- Imaginary device
  - Well defined set of characteristics
- Both sides generate data and control signals in native language but translates them to NVT form
  - The sending side translates native data and control signals into NVT form before sending out
  - the receiving side gets the NVT data and signals and translates into its native form
NVT Operation

- Accommodating heterogeneity

Converting client system format into NVT format

Converting NVT format into server system format

Converting NVT format into client system format

Converting server system format into NVT format

TCP connection across Internet

Client System format

Server System format

NVT format

TELNET client and server convert between native format and NVT format
NVT Format

- NVT use two sets of characters: data and control.
- Standard format of data characters:
  - Standard ASCII

- Standard format of control characters:

- Standard format has been extended to support the data characters to be transmitted as 8 bit bytes with most significant bit set to 1.
TELNET Operations
TELNET Operations

Connection management
- Connection request, establish and terminate
- Telnet uses TCP (port 23) by default

Negotiation
- To determine mutually agreeable set of characteristics and options

Exchange of control information (e.g. end of line), commands and transfer of data between two correspondents

A typical telnet session is exchange of data between terminal and host
- Multiple rounds
- Not only for accessing remote accounts; was also used for interactive system
  - Try “telnet bbs.byr.cn”
TELNET Protocol
Related RFCs

- **Basic protocol**
  - RFC854: Telnet Protocol Specification

- **Options**
  - RFC855: Telnet Option Specifications
  - RFC856: Telnet Binary Transmission
  - RFC857: Telnet Echo Option
  - RFC858: Telnet Suppress Go Ahead Option
  - RFC859: Telnet Status Option
  - …
Some Features

- **TCP connection**: directed toward port 23 of the server being asked to perform a service.
- Data and control *multiplexed* over the same connection.
- **NVT** - representation of a *generic* terminal.
- **Negotiated options** - provides a standard language for communication of terminal control functions.
TELNET Protocol

- Transmission of data
- Standard representation of control functions
Transmission Of Data (1)

Data path from the user’s keyboard to the remote system

- Client reads from terminal
- TELNET client
- Client sends to server (NVT)
- Server receives from client (NVT)
- TELNET server
- Server sends to pseudo terminal
- User’s keyboard & display
- Operating system
- TCP/IP internet
- Operating system
Transmission Of Data (2)

- Data sent half-duplex
  - Terminal-to-process, newline signifies end of user input
  - Process-to-terminal, control signal Go Ahead(GA) is used
- Underlying TCP full duplex
  - Control signals sent any time regardless of current data direction
- Data sent as stream of 8-bit bytes
  - No other formatting
- Control signals and other non-data information sent as Telnet commands
  - Byte strings embedded in data stream
  - User control signals, commands between Telnet processes as part of protocol and option negotiation and subnegotiation
Control Functions (1)

- TELNET includes support for a series of control functions commonly supported by servers
- This provides a uniform mechanism for communication of (the supported) control functions
- You can imagine them as some extra virtual keys in the NVT keyboard
Control Functions (2)

- **Interrupt Process (IP)**
  - Suspend/interrupt/abort/terminate process

- **Abort Output (AO)**
  - allow a process, which is generating output, to run to completion but without sending the output to the user's terminal

- **Are You There (AYT)**
  - check to see if system is still running

- **Erase Character (EC)**
  - delete last character sent
  - typically used to edit keyboard input

- **Erase Line (EL)**
  - delete all input in current line
  - typically used to edit keyboard input
# Control Functions (3) – delivery

<table>
<thead>
<tr>
<th>Command</th>
<th>Decimal Codes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAC</td>
<td>255</td>
<td>Interpret next octet as command</td>
</tr>
<tr>
<td>DONT</td>
<td>254</td>
<td>Denial of request to perform specific option</td>
</tr>
<tr>
<td>DO</td>
<td>253</td>
<td>Approval to allow specific option</td>
</tr>
<tr>
<td>WONT</td>
<td>252</td>
<td>Refusal to perform specific option</td>
</tr>
<tr>
<td>WILL</td>
<td>251</td>
<td>Agreement to perform specific option</td>
</tr>
<tr>
<td>SB</td>
<td>250</td>
<td>Start of option subnegotiation</td>
</tr>
<tr>
<td>GA</td>
<td>249</td>
<td>Go ahead</td>
</tr>
<tr>
<td>EL</td>
<td>248</td>
<td>Erase line</td>
</tr>
<tr>
<td>EC</td>
<td>247</td>
<td>Erase character</td>
</tr>
<tr>
<td>AYT</td>
<td>246</td>
<td>Are you there</td>
</tr>
<tr>
<td>AO</td>
<td>245</td>
<td>Abort output</td>
</tr>
<tr>
<td>IP</td>
<td>244</td>
<td>Interrupt process</td>
</tr>
<tr>
<td>BRK</td>
<td>243</td>
<td>Break</td>
</tr>
<tr>
<td>DMARK</td>
<td>242</td>
<td>Data mark</td>
</tr>
<tr>
<td>NOP</td>
<td>241</td>
<td>No operation</td>
</tr>
<tr>
<td>SE</td>
<td>240</td>
<td>End of subnegotiation</td>
</tr>
<tr>
<td>EOR</td>
<td>239</td>
<td>End of record</td>
</tr>
</tbody>
</table>
Control Functions (4) – IAC

- **TELNET command structure**
  - at least a two byte sequence: the IAC (Interpret as Command) escape character followed by the code for the command
  - The IAC code is **255**
    - If a 255 is sent as data - it must be followed by another 255
  - Looking for a command
    - Each receiver must look at each byte that arrives and look for an IAC
    - If IAC is found and the next byte is “IAC” - a single data byte (value 255) is presented to the application/terminal
    - If IAC is followed by any other code - the TELNET layer interprets this as a command
Control Functions (5) – DO, DONT, WILL, WONT

- Used for options negotiation
- Examples

<table>
<thead>
<tr>
<th>Sender</th>
<th>Receiver</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>WILL →</td>
<td>← DO</td>
<td>Sender wants to active a option, and receiver agrees</td>
</tr>
<tr>
<td>WILL →</td>
<td>← DON’T</td>
<td>Sender wants to active a option, and receiver refuses</td>
</tr>
<tr>
<td>DO →</td>
<td>← WILL</td>
<td>Sender wants receiver to active a option, and receiver agrees</td>
</tr>
<tr>
<td>DO →</td>
<td>← WONT</td>
<td>Sender wants receiver to active a option, and receiver refuses</td>
</tr>
</tbody>
</table>
TELNET Options Negotiation
Motivations

- All NVTs support a minimal set of capabilities.
- Some terminals have more capabilities than the minimal set.
- The two endpoints negotiate a set of mutually acceptable options (character set, echo mode, etc).
- The set of options is not part of the TELNET protocol, so that new terminal features can be incorporated without changing the TELNET protocol.
Option Examples

- echo modes
  - Keyboard input be echoed on the terminal side or not
- Line mode vs. character mode
  - One line or one character per transmission
- character set (EBCDIC vs. ASCII)
  - EBCDIC - Extended Binary-Coded Decimal Interchange Code
  - ASCII - American Standard Code for Information Interchange
Options Negotiation

- Each option is assigned a byte value
- The DO, DONT, WILL, and WONT commands are used to negotiate options
- Options negotiation is symmetric
- Steps must be taken to avoid option processing loops
- Subnegotiations are used when more information is needed, such as when negotiating terminal type, window size, etc
Example: Negotiation of Echo Option

Client

Do enable the echo option

```
IAC
DO
ECHO
```

Server

```
IAC
WILL
ECHO
```

I will enable the echo option
### TELNET Options List (1)

<table>
<thead>
<tr>
<th>Option</th>
<th>Name</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Binary Transmission</td>
<td>RFC 856</td>
</tr>
<tr>
<td>1</td>
<td>Echo</td>
<td>RFC 857</td>
</tr>
<tr>
<td>2</td>
<td>Reconnection</td>
<td>NIC 50005</td>
</tr>
<tr>
<td>3</td>
<td>Suppress Go Ahead</td>
<td>RFC 858</td>
</tr>
<tr>
<td>4</td>
<td>Approx Message Size Negotiation</td>
<td>ETHERNET</td>
</tr>
<tr>
<td>5</td>
<td>Status</td>
<td>RFC 859</td>
</tr>
<tr>
<td>6</td>
<td>Timing Mark</td>
<td>RFC 860</td>
</tr>
<tr>
<td>7</td>
<td>Remote Controlled Trans and Echo</td>
<td>RFC 726</td>
</tr>
<tr>
<td>8</td>
<td>Output Line Width</td>
<td>NIC 50005</td>
</tr>
<tr>
<td>9</td>
<td>Output Page Size</td>
<td>NIC 50005</td>
</tr>
<tr>
<td>10</td>
<td>Output Carriage-Return Disposition</td>
<td>RFC 652</td>
</tr>
<tr>
<td>11</td>
<td>Output Horizontal Tab Stops</td>
<td>RFC 653</td>
</tr>
<tr>
<td>12</td>
<td>Output Horizontal Tab Disposition</td>
<td>RFC 654</td>
</tr>
<tr>
<td>13</td>
<td>Output Formfeed Disposition</td>
<td>RFC 655</td>
</tr>
<tr>
<td>14</td>
<td>Output Vertical Tabstops</td>
<td>RFC 656</td>
</tr>
<tr>
<td>15</td>
<td>Output Vertical Tab Disposition</td>
<td>RFC 657</td>
</tr>
<tr>
<td>16</td>
<td>Output Linefeed Disposition</td>
<td>RFC 658</td>
</tr>
<tr>
<td>17</td>
<td>Extended ASCII</td>
<td>RFC 698</td>
</tr>
<tr>
<td>18</td>
<td>Logout</td>
<td>RFC 727</td>
</tr>
<tr>
<td>19</td>
<td>Byte Macro</td>
<td>RFC 735</td>
</tr>
</tbody>
</table>
**TELNET Options List (2)**

<table>
<thead>
<tr>
<th>Option</th>
<th>Name</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Data Entry Terminal</td>
<td>RFC 1043, RFC 732</td>
</tr>
<tr>
<td>21</td>
<td>SUPDUP</td>
<td>RFC 736, RFC 734</td>
</tr>
<tr>
<td>22</td>
<td>SUPDUP Output</td>
<td>RFC 749</td>
</tr>
<tr>
<td>23</td>
<td>Send Location</td>
<td>RFC 779</td>
</tr>
<tr>
<td>24</td>
<td>Terminal Type</td>
<td>RFC 1091</td>
</tr>
<tr>
<td>25</td>
<td>End of Record</td>
<td>RFC 885</td>
</tr>
<tr>
<td>26</td>
<td>TACACS User Identification</td>
<td>RFC 927</td>
</tr>
<tr>
<td>27</td>
<td>Output Marking</td>
<td>RFC 933</td>
</tr>
<tr>
<td>28</td>
<td>Terminal Location Number</td>
<td>RFC 946</td>
</tr>
<tr>
<td>29</td>
<td>Telnet 3270 Regime</td>
<td>RFC 1041</td>
</tr>
<tr>
<td>30</td>
<td>X.3 PAD</td>
<td>RFC 1053</td>
</tr>
<tr>
<td>31</td>
<td>Negotiate About Window Size</td>
<td>RFC 1073</td>
</tr>
<tr>
<td>32</td>
<td>Terminal Speed</td>
<td>RFC 1079</td>
</tr>
<tr>
<td>33</td>
<td>Remote Flow Control</td>
<td>RFC 1372</td>
</tr>
<tr>
<td>34</td>
<td>Linemode</td>
<td>RFC 1184</td>
</tr>
<tr>
<td>35</td>
<td>X Display Location</td>
<td>RFC 1096</td>
</tr>
<tr>
<td>36</td>
<td>Environment Option</td>
<td>RFC 1408</td>
</tr>
<tr>
<td>37</td>
<td>Authentication Option</td>
<td>RFC 2941</td>
</tr>
<tr>
<td>38</td>
<td>Encryption Option</td>
<td>RFC 2946</td>
</tr>
</tbody>
</table>
A Telnet Session Example (1)

C:\Documents and Settings\Administrator> telnet 192.168.1.253
Red Hat Enterprise Linux AS release 4 <Nahant Update 1>
Kernel 2.6.9-11.Elsmp on an i686
Login: shiyan
Password:
Last login: Sun Nov 11 17:48:30 from 192.168.1.168
[shiyan@localhost ~]$
Some options negotiated firstly
Suboption about the terminal type
ANSI / DEC / IBM3270 / ...

IAC SB Terminal Type
IAC SE

0000 00 13 72 4f 9d 3a 00 15 f2 92 f0 00 45 00 .ro......E.
0010 00 32 3a 0a 4a 00 80 0a 6b c0 a8 01 a8 c0 a8 .2@...k...
0020 01 fd 04 cd 00 17 8d 0c b3 f0 64 98 7a 0c 50 18 ...
0030 fa e9 5f 9a 00 ff fa 18 00 41 4e 53 49 ff f0 ...ANSI
Some prompt information given by the server

Data and control multiplexed over the same connection
When I typed in login ID “shiyan”
When I pressed ENTER
Summary (1) – usages of telnet

- Use Internet accounts you may have on remote computers
  - you need an account (login ID) and password on the remote computer to permit access
- Use free services accessible with telnet, e.g.
  - library catalogues
  - databases
  - BBS (Bulletin Board System)
  - Router/switch configuration
Summary (2) – Disadvantages of telnet

- Poor user interface
  - Based on dumb terminal
  - Text-only display
  - Monochrome
    - One color for text, one for background
  - Have to type command-line commands
    - Often have complex syntax
  - Not very secure, SSH made enhancement
    - TELNET does not encrypt any data sent over the connection (including passwords)
Other Remote Access Technologies
Other Remote Access Technologies

- **Remote login** in text-based system
  - telnet
  - SSH
  - Rlogin

- **Remote desktop** in windowing system
  - VNC (Virtual Network Computing)
  - RDP (Remote Desktop Protocol)
SSH (1) – brief information

- Secure Shell
- Command line terminal connection tool
- All traffic encrypted
- Both ends authenticate themselves to the other end
- Ability to carry and encrypt non-terminal traffic
- Private key kept on client, public key stored on server
- Now, it is an IETF standard
  - RFC4251, The Secure Shell (SSH) Protocol Architecture
SSH (2) – two enhancements of telnet

- Providing secure communications
- Providing users with the ability to perform additional, independent data transfer over the same connection that is used for remote login
SSH (3) – three major mechanisms

- A **transport layer protocol** that provides server authentication, data confidentiality, and data integrity with perfect forward secrecy

- A **user authentication protocol** that authenticates the user to the server

- A **connection protocol** that multiplexes multiple logical communications channels over a single underlying SSH connection
  - Port forwarding, could be used as a secure tunnel
SSH (4) – tools

PuTTY
### Wireshark Capture of SSH Traffic

The following table shows a capture of SSH traffic using Wireshark.

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.009699</td>
<td>192.168.1.253</td>
<td>192.168.1.168</td>
<td>SSH</td>
<td>Server Protocol: SSH-1.99-OpenSSH_3.9p1</td>
</tr>
<tr>
<td>5</td>
<td>0.031789</td>
<td>192.168.1.168</td>
<td>192.168.1.253</td>
<td>SSH</td>
<td>Client Protocol: SSH-1.5-PuTTY-Release-0.53b</td>
</tr>
<tr>
<td>7</td>
<td>0.033267</td>
<td>192.168.1.253</td>
<td>192.168.1.168</td>
<td>SSH</td>
<td>Client Protocol: SSH-1.5-PuTTY-Release-0.53b</td>
</tr>
<tr>
<td>8</td>
<td>0.074425</td>
<td>192.168.1.168</td>
<td>192.168.1.253</td>
<td>SSHV1</td>
<td>Server: Public Key</td>
</tr>
<tr>
<td>9</td>
<td>0.085057</td>
<td>192.168.1.168</td>
<td>192.168.1.253</td>
<td>SSHV1</td>
<td>Client: Session Key</td>
</tr>
<tr>
<td>10</td>
<td>2.643195</td>
<td>192.168.1.168</td>
<td>192.168.1.253</td>
<td>SSHV1</td>
<td>Client: Encrypted packet len=5</td>
</tr>
<tr>
<td>11</td>
<td>2.643750</td>
<td>192.168.1.168</td>
<td>192.168.1.253</td>
<td>SSHV1</td>
<td>Client: Encrypted packet len=5</td>
</tr>
<tr>
<td>12</td>
<td>4.451871</td>
<td>192.168.1.168</td>
<td>192.168.1.253</td>
<td>SSHV1</td>
<td>Client: Encrypted packet len=24</td>
</tr>
<tr>
<td>14</td>
<td>4.472525</td>
<td>192.168.1.168</td>
<td>192.168.1.253</td>
<td>SSHV1</td>
<td>Client: Encrypted packet len=31</td>
</tr>
<tr>
<td>16</td>
<td>4.477296</td>
<td>192.168.1.168</td>
<td>192.168.1.253</td>
<td>SSHV1</td>
<td>Client: Encrypted packet len=5</td>
</tr>
<tr>
<td>18</td>
<td>4.528057</td>
<td>192.168.1.168</td>
<td>192.168.1.253</td>
<td>SSHV1</td>
<td>Client: Encrypted packet len=54</td>
</tr>
<tr>
<td>19</td>
<td>4.528057</td>
<td>192.168.1.168</td>
<td>192.168.1.253</td>
<td>SSHV1</td>
<td>Client: Encrypted packet len=10</td>
</tr>
<tr>
<td>20</td>
<td>4.528057</td>
<td>192.168.1.168</td>
<td>192.168.1.253</td>
<td>SSHV1</td>
<td>Client: Encrypted packet len=10</td>
</tr>
<tr>
<td>21</td>
<td>4.528057</td>
<td>192.168.1.168</td>
<td>192.168.1.253</td>
<td>SSHV1</td>
<td>Client: Encrypted packet len=10</td>
</tr>
<tr>
<td>22</td>
<td>4.528057</td>
<td>192.168.1.168</td>
<td>192.168.1.253</td>
<td>SSHV1</td>
<td>Client: Encrypted packet len=10</td>
</tr>
</tbody>
</table>

---

**Frame 11 (74 bytes on wire, 74 bytes captured)**


**SSH Protocol**
- **SSH Version 1**
  - Packet Length: 15
  - Padding Length: 1
  - Payload: 0CE5B0115E08CCFB2924936DE8C777
Other Ways Of Remote Access

- Except telnet, there are other ways
  - `rlogin` family utility
  - VNC (Virtual network computing)
  - RDP (Remote Desktop Protocol)
- Comparison with Telnet
Helpful URLs

- **RFCs**
  - http://www.ietf.org/

- **Useful utilities**

- **About telnet**

- **About SSH**
  - http://www.ssh.com
  - http://www.openssh.org

- **About realVNC**
  - http://www.realvnc.com/
### Abbreviations of Week 6-9

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBS</td>
<td>Bulletin Board System</td>
</tr>
<tr>
<td>BOOTP</td>
<td>BOOTstrap Protocol</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>FQDN</td>
<td>Fully Qualified Domain Name</td>
</tr>
<tr>
<td>IAC</td>
<td>Interpret As Command</td>
</tr>
<tr>
<td>MSC</td>
<td>Message Sequence Chart</td>
</tr>
<tr>
<td>NVT</td>
<td>Network Virtual Terminal</td>
</tr>
<tr>
<td>SSH</td>
<td>Secure SHell</td>
</tr>
<tr>
<td>STD</td>
<td>State Transition Diagram</td>
</tr>
<tr>
<td>TLD</td>
<td>Top Level Domain</td>
</tr>
<tr>
<td>TLV</td>
<td>Type-Length-Value</td>
</tr>
<tr>
<td>TTL</td>
<td>Time To Live</td>
</tr>
</tbody>
</table>