TNE30019/TNE80014 – Unix for Telecommunications

Remote Access to a Unix System

Dr. Jason But

Swinburne University

Outline

Once upon a time...

Remote access to machines always important feature of Unix

- Run programs/applications remotely
- Remote maintenance
- Network development and testing
- Router development – remember that routers originally were just computers with multiple network interfaces

Unix – Remote Access

- GUIs were not originally used
- All access was command line interface (CLI)
  - Via terminal
  - Over network
- Terminal device `/dev/tty` is ideal abstraction
- Concept of terminal was extended to run over network
- Network connection is created and `tty` device is attached to remote connection

Telnet

First remote access client/server implementation

Telnet – Client and Server

Telnet client

- Uses TCP – reliable in-order delivery
- Behaves like terminal
  - Redirects all input from **stdin** through TCP socket
  - Redirects all input received from TCP socket to **stdout**
- Useful for any text based protocol

Telnet server – `telnetd`

- Configuration file `/etc/telnetd.conf`
- Opens socket on telnet port (23) and accepts connections
  - Creates new virtual terminal
  - Ties **stdin** on terminal to data read from socket
  - Ties **stdout** on terminal to output to socket
  - Launches `login` on terminal device
- Remote user interacts with login and possibly shell
- Supports multiple concurrent connections

Telnet – Security Issues

Everything is plain text

Network sniffing tools can detect

- Username/Password
- What programs are run
- What data is sent/received

- Originally not an issue
  - All network users were involved in building network
  - No malicious users

Typically today Unix...

- Does not have `telnetd` installed; OR
- Has `telnetd` installed but not started at system boot; OR
- System administrators disable `telnetd`

- Telnet client is a completely different story...

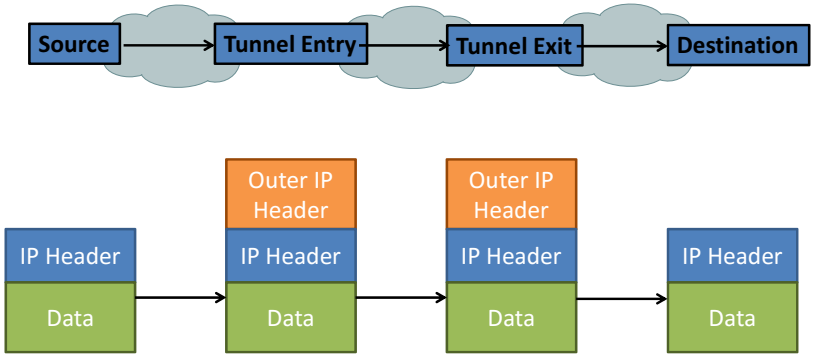
Improving Security Through Tunneling

What is a tunnel?

- IP Packets are encapsulated within IP packets
- Create virtual point-to-point link over the network
- Outside packet is routed over network as normal
- Inside packet is encapsulated at tunnel entry and decapsulated at tunnel exit
- Decapsulated packet is routed normally to destination

- IPSec is one protocol to provide secure tunnel through public IP network
- Multiple networks connected by secure tunnels are called Virtual Private Network (VPN)

Improving Security Through Tunneling



IPSec

What is IPSec?

- Needs to be configured at tunnel endpoints
- Secure session is negotiated between endpoints (session key)
- IPSec provides either
  - Authenticated transfer: guaranty integrity and data origin authentication of packets
  - Secure transfer: guaranty authenticity, integrity and confidentiality of packets

- If IPSec tunnels packets to firewalled trusted network
  - Using Telnet to server within trusted network is OK
  - Telnet session is secured by VPN
- Many different products out there
  - Google for "IPSec, VPN, Unix"

SSH – Secure Shell Protocol

- Open protocol[1] to provide secure shell
- Generic modern replacement for Telnet

Uses standard cryptographic techniques

- Certificates to verify identity of remote system
- Public key encryption to exchange session keys
- Private key encryption to secure communications

[1] `http://www.openssh.com`

SSH – Secure Shell

Basic Protocol

- TCP session created between two hosts
- Server sends certificate
  - Ties server's public key to its identity
  - Cryptographically signed by trusted authority
  - Checked against copy received from previous session
  - If first session, user queried about accepting certificate
- Client and server negotiate common cipher to use
- Random session key is generated using public key encryption
- Further communication is encrypted using private session key
- `login` is executed which subsequently launches shell

- SSH can also be told to launch different shell

SSH – Configuring and Starting SSH Server

Configuration

- All configuration files live in `/etc/ssh`
- Server (daemon) configuration: `/etc/ssh/sshd_config`
- See man `sshd_config`

`sshd` – SSH Daemon

- Typically started automatically by `rc`
- First time started requests random keyboard input to generate public certificates

- When configuration file is changed, daemon must be restarted
  - Kill process ID and restart
  - Send OS signal SIGHUP (Hang UP) to process ID

SSH – Configuring SSH Client

`/etc/ssh/ssh_config` System default client configuration

`$(HOME)/.ssh/config` Override configuration for user

Command Line Options Overide all configuration for this session

`$(HOME)/.ssh/known_hosts` Public keys of known hosts

`$(HOME)/.ssh/authorized_keys` Public keys of authorised hosts

Available SSH Clients

- `ssh` – **S**ecure **SH**ell
- `scp` – **S**ecure **CoP**y
  - Transfers files over secure session
  - Can recursively copy directories
- `sftp` – **S**ecure **FTP**
  - Interactively transfers files over secure session
  - FTP like interactive shell to allow navigation of remote site
- All use SSH as underlying secure protocol

SSH – Port Forwarding

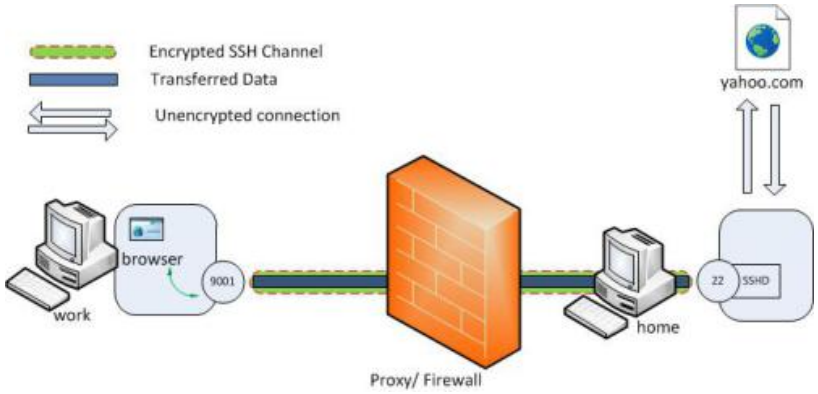
- Port on local host can be forwarded to port on remote host
- Port on remote host can be forwarded to port on local host

Mini-VPN

- Access to single (or small set) of applications is granted via **SSH** tunnel
- Access single service that would otherwise not be available

SSH – Local Port Forwarding

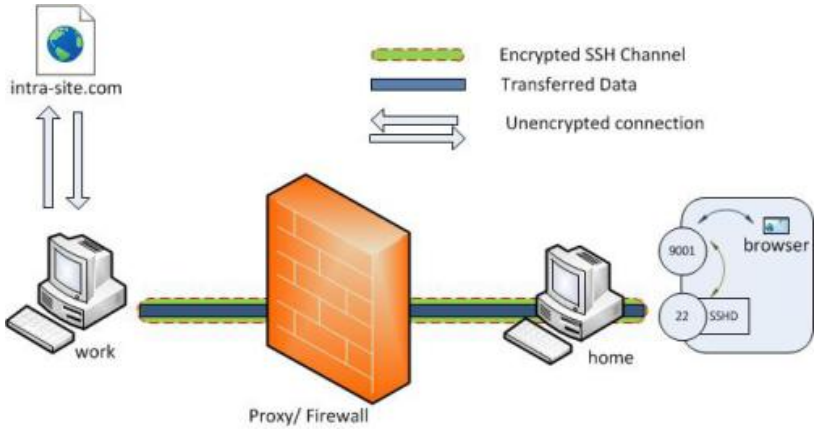
Work machine: ssh -L 9001:yahoo.com:80 home



http://chamibuddhika.wordpress.com/2012/03/21/ssh-tunnelling-explained

SSH – Reverse Port Forwarding

Work machine: ssh -R 9001:intra-site.com:80 home



http://chamibuddhika.wordpress.com/2012/03/21/ssh-tunnelling-explained

X Server – Running Remote Applications

- `DISPLAY` variable or command line parameter tells X application which host is X Server
- Insecure, no access control
- In old times simple access control with `xhost` and `xauth`

`xhost` program

- `xhost +<host>` allows anybody on `<host>` to access X server

`xauth` program

- Share magic cookie between client and server
- Client must present cookie to display

- Inconvenient to set up and still insecure, especially un-encrypted communications between client and server

X Server – X Forwarding over SSH

- Easiest and most secure solution
- New display is created and tunneled through **SSH** session
- Environment variables (e.g. `DISPLAY`) automatically set
- However, we need to configure this

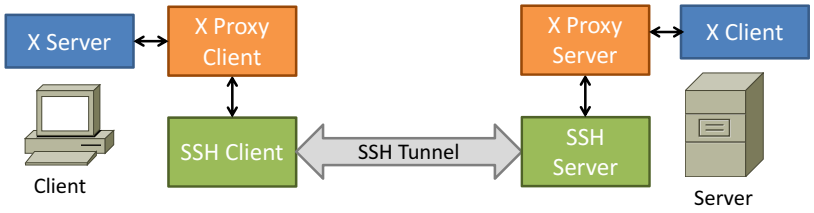
sshd Configuration – `/etc/ssh/sshd_config`

- Set `X11Forwarding Yes`

Client Configuration – `/etc/ssh/ssh_config` (`~/.ssh/config`)

- Set `ForwardX11 Yes`
- Optionally set `ForwardX11Trusted Yes`
- Or run `ssh -XY`

X Server – X Forwarding over SSH



- SSH Server starts X Proxy Server (sets DISPLAY to X proxy)
- SSH Client starts X Proxy Client (DISPLAY set to X Server)

X Server – Issues

- X protocol is a bit of a bandwidth hog
  - Send basic operations, e.g. open/position/resize windows, ...
  - Some applications or underlying libraries are very verbose
  - Modern applications more graphically intensive
  - SSH supports compression, but still not good enough
- High latency (lag) with X
  - Due to synchronous operations
  - SSH encryption/compression adds more latency
- X not good over low bandwidth or high latency links
- If network outage or X server down we have problems
  - Network outage will terminate applications
  - Can't suspend and resume user machine (it's X server!)

VNC – Virtual Network Computing

- VNC Protocol developed for **platform independent** remote computer access[2] – `http://www.realvnc.com/`

Based on simple idea

1. Client connects to remote server
2. Authentication occurs
3. Negotiation of different transfer methods
4. Server sends complete framebuffer bitmap
5. Client periodically requests updates to framebuffer
6. Server sends bitmaps only for **changed** parts of framebuffer
7. Different encoding methods, e.g. raw, compressed

[2] T Richardson, Q Stafford-Fraser, K Wood, A Hopper, "Virtual Network Computing", IEEE Internet Computing, Vol. 2/No. 1, Jan/Feb 1998

VNC – Virtual Network Computing

- Better use of network resources
  - More friendly to low bandwidth connections
  - Client can adjust update rate based on network bandwidth
- Session runs on server
  - Can resume after network outage
  - Can disconnect and reconnect client (display)
- Clients and servers available for many platforms
- Client is thin
  - Java client to embed in browser
  - Can be run on small devices