– General Linux 2 –

Set up secure shell (OpenSSH) []

(Linux Professional Institute Certification)

[a]

```
   .~.       Based on a talk by Gus Lees
   /V\
  // \\      geoffrey robertson
  @._.@      geoffrey@zip.com.au
```

`$Id: gl2.113.7.slides.tex,v 1.4 2003/10/25 04:27:27 anguslees Exp $`

Set up Secure Shell (OpenSSH)

Objective

The candidate should be able to obtain and configure OpenSSH. This objective includes basic OpenSSH installation and troubleshooting, as well as configuring sshd to start at system boot.

Set up Secure Shell (OpenSSH)

Key files, terms, and utilities

```
/etc/hosts.allow
/etc/hosts.deny
/etc/nologin
/etc/ssh/sshd_config
/etc/ssh_known_hosts
/etc/sshrc
sshd
ssh-keygen
```

Set up Secure Shell (OpenSSH)

Resources of Interest

TBA

“Secure SHell”

A functional replacement of the ancient `rsh` command, except with encryption and authentication.

Versions

Commercial SSH Finnish company. Original authors of SSH.

OpenSSH Split from last free version of commercial SSH. Development led by OpenBSD team.

Draft “secsh” RFC.

Alternative implementations exist (Putty, Net::SSH::Perl, etc)

Commands

ssh Run a shell command on a remote host

sshd SSH server daemon

scp Copy files using SSH

sftp An ftp-like interface into `scp`

ssh-keygen Generate an SSH key pair

ssh-agent, ssh-add SSH key forwarding

Commands

ssh

```
ssh [options] host [command]
```

Run a shell command on a remote host.

Acts like a normal shell command. ie: STDIN, STDOUT work as normal.

Without a *command*, ssh runs an interactive login.

Commands

scp

```
scp user@host:path/file user2@host2:path/file2
```

Copy a file over ssh.

user defaults to current login, *user@host* maybe omitted for local files, *path* is relative to $HOME

Commands

sftp

```
sftp user@host:path
```

`ftp`-like command line interface to scp.

Only provided with more recent ssh versions.

Advanced Usage

Remember that STDIN and STDOUT still work as normal (unlike `telnet`):

```
ssh remote tar zcf - /remotepath > localfile.tar.gz
```

Advanced Usage

X Forwarding

```
ssh -X host
```

Login to *host* and “forward” X11 connections back to the local Xserver.

A “fake” $DISPLAY and `xauth` environment are created, and the X11 data is passed back over the same SSH connection.

Advanced Usage

X Forwarding *ctd...*

Forwarding X over SSH is secure and easy, but slower than not doing it.

On a local LAN, the encryption is probably unnecessary—use normal X methods such as rstart instead (rstart can use ssh for authentication anyway).

Specialised X11 caching methods (eg: LBX) can get better performance than ssh compression over slow links.

There are concerns over connecting to a hostile remote machine and forwarding X back again, so don't forward X by default. A hostile remote site may forward damaging commands back down the link to your X server (ie your screen and keyboard).

Advanced Usage

Port Forwarding

Arbitrary ports can also be forwarded over the SSH connection, to add security to other protocols (or bypass poor firewall policies. . .)

```
# .fetchmailrc example
poll localhost protocol pop3 port 11110:
  preconnect "ssh -C -f user@host.com \
    -L 11110:host.com:110 sleep 10"
```

Advanced Usage

SSH Keys

Public key authentication. More secure alternative to password login.

Generate a public/private “key pair” with `ssh-keygen`.

Keep the private key secret.

Append the public key into your (remote) `~/.ssh/authorized_keys` to allow access.

```
$ cat identity.pub >> ~/.ssh/authorized_keys ↩
```

More powerful automation (scripting) possibilities.

Advanced Usage

SSH Authentication Agent

`ssh-agent` allows key information to be “forwarded” between its child processes—even across nested ssh sessions.

Start `ssh-agent` in your X-session or login scripts, and run `ssh-add` to add keys.

`ssh-askpass` is (basically) an X11 version of `ssh-add`.

Advanced Usage

SSH from win32

Putty Includes command line “pscp.exe” scp clone too.

`http://www.chiark.greenend.org.uk/˜sgtatham/putty/`

Winscp Graphical SCP client.

`http://winscp.vse.cz/eng`

Advanced Usage

Other “frontends”

KDE kio_fish Provides `ssh://` konquerer paths.

tramp.el Transparent access to remote files for emacs.

rsh-compatible Anything that can use rsh (eg: CVS)

The End