Using ssh in a mixed OS environment

SSH, the Secure Shell, is available as a native tool and often bundled with distributions of many UNIX-like operating systems, including Linux and Mac OS X, and can be grafted on top of Windows operating systems. The primary function of ssh is to establish a secure and encrypted tunnel between two machines. This tunnel can allow machines to connect through open public networks without disclosing the information contained within the tunnel.

Using the ssh tunnel, additional services can be rerouted through the tunnel. Most ssh implementations include SCP, the secure copy program, and may include additional tools for the generation of keys, which can allow for automated batch connections (rather than interactive sessions that require a password). In addition, services on any port can be redirected through the tunnel.

SSH requires that software be running at both ends of the connection: an ssh server runs on one machine, and ssh client software runs on the other. The connection is initiated by client.

Linux Servers

The SSH daemon, sshd, can be installed from all Linux distributions. Installation and configuration vary with the distribution. A good white paper on basic install is available at http://www.hentzenwerke.com/wp/remoteaccessviassh.htm.A great deal more information, including source code, documentation and pointers to many more resources, can be found at http://www.openssh.org.

Windows Servers

Serving ssh on Windows machines is generally not recommended, as Windows machine security is more problematic. However, an ssh server is available as part of the CygWin set of tools for a POSIX interfaces atop Windows. See http://www.cygwin.com/

Windows Clients

Windows client tools: http://www.openssh.org/windows.html
PuTTY is great for a simple terminal, and will also let you set up and save complex settings for items like port forwarded. Free.
WinSCP is a great graphical tool for SCP (secure file copy) with a two-pane interface to copy or move files to and from remote machines.

Operation

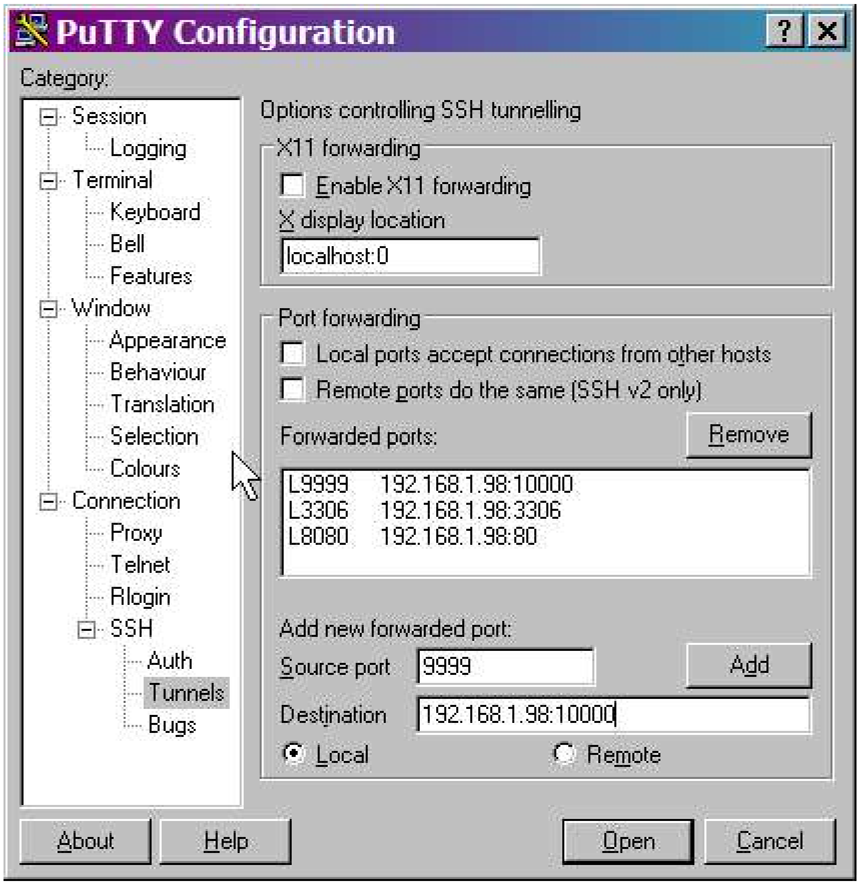
On Linux/OS X machines, issue a command like:

```
ssh -p2233 yourname@example.com
```

to attempt to log onto the example.com server as user 'yourname' via port 2233 (It's generally a good practice to change the default port 22 configuration for ssh, just to avoid the most basic of automated scripts).

```
ssh -p2222 -L 9999:192.168.1.99:10000 yourname@example.com
```

uses the same login as before, but also redirects to the local port 9999 the traffic that would normally be seen on port 10000 of the remote host machine at IP address 192.168.1.99 – note that this may or may not be the ssh server at example.com, but may in fact be relaying something from another machine in example.com's local area network. This example forwards port 10000, often the WebMin server (http://www.webmin.com) so that it can be accessed remotely and securely by the ssh client machine at http://localhost:9999. Figure 1 illustrates a similar configuration using the Windows client PuTTY.



Options controlling SSH tunnelling

X11 forwarding

□ Enable X11 forwarding

X display location: localhost:0

Port forwarding

□ Local ports accept connections from other hosts

□ Remote ports do the same (SSH v2 only)

Forwarded ports:

```
L9999    192.168.1.98:10000
L3306    192.168.1.98:3306
L8080    192.168.1.98:80
```

Add new forwarded port:

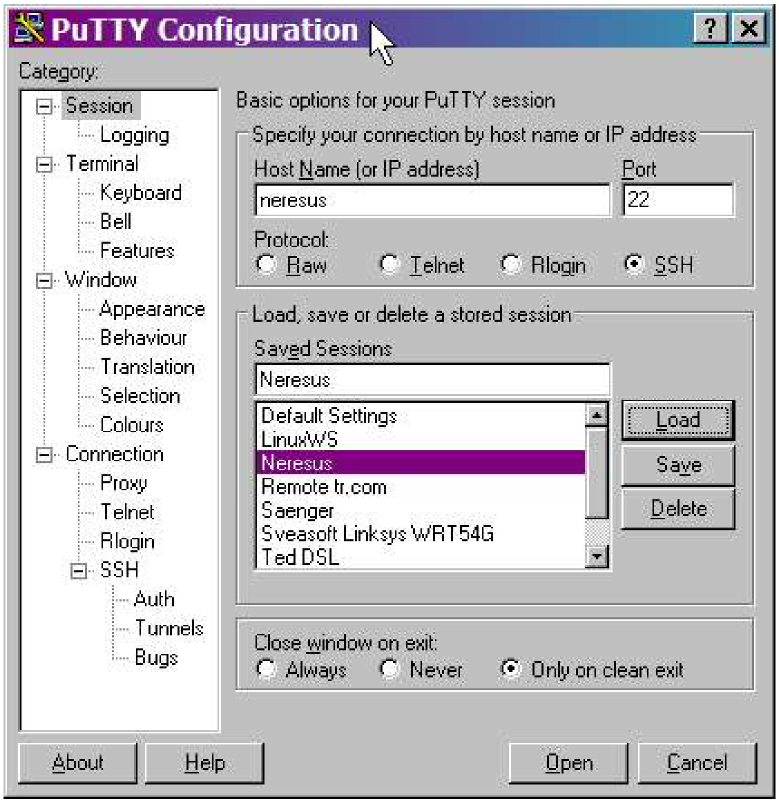
Source port: 9999

Destination: 192.168.1.98:10000

● Local ○ Remote

Figure 1: Sample PuTTY configuration with three ports forwarded to local machine. Textboxes are filled in to show how the first entry was created.

To start an ssh session using PuTTY, start the Putty executable, and specify the IP address or friendly name (example.com) of the server and the port it is running on, and select “Open.” If this is a server you plan to regularly connect to, you can use the “Load” and “Save” functions to preserve your settings. Note the the settings are stored to the Registry for the current user, but do not include login information. See Figure 2 for a typical basic login screen.



Basic options for your PuTTY session

Specify your connection by host name or IP address

Host Name (or IP address): neresus

Port: 22

Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

Load, save or delete a stored session

Saved Sessions: Neresus

Default Settings
LinuxWS
Neresus
Remote tr.com
Saenger
Sveasoft Linksys WRT54G
Ted DSL

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 2: Loading a previously saved configuration, these settings will connection to a host named 'neresus' on port 22 using ssh.