**Local Port Forwarding**

ssh -L sourcePort:forwardToHost:onPort connectToHost

connect to *connectToHost*, and forward all connections to the local *sourcePort* to port *onPort* on host called *forwardToHost*, which can be reached from the *connectToHost* machine.

Example 1:

ssh -v -L 43022:localhost:80 web1 -N &

Forward local port 43022 to port 80 on specified remote host (web1) then can run “http://localhost:43022” from a browser on your localhost.

or: nc -v localhost 43022 -z

or: curl -Iv localhost:43022

-N means no shell, just forward the port,

this assumes SAME user ID on local and remote host, can also specify the user on the remote host:

ssh -L 43022:localhost:80 jimmy@web1 -N

#you will get login shell without the -N

ssh -l jk24445 -L 43022:localhost:80 phlutlxsand001

Example 2:

ssh -L 8080:www.ubuntuforums.org:80 myhost

myhost is the name of your machine. Now can connect to the remote host via <http://localhost:8080/>

or ‘curl myhost:8080’

Example 3:

ssh -L 43022:sand001:80 jmp001 -N &

Same as example one but connect to jump server first then forward connections on localhost to sand001:80

Example 4:

ssh -L **localhost**:43022:sand001:80 jmp001 -N &

Supply a bind address (localhost): local port is bound in accordance with the GatewayPorts setting. An explicit bind\_address may be used to bind the connection to a specific address

**Remote Port Forwarding**

ssh -R sourcePort:forwardToHost:onPort connectToHost

Connect to *connectToHost*, and forward all connections to the **remote** *sourcePort* to port *onPor*t on the machine called *forwardToHost*, which can be reached from your local machine.

Example 1:

[jk24445@phlutlxsand001 ~]#ssh -R 9000:phlutlxsand001:80 jk24445@phlprlxjmp001

Forward port 80 on localhost (sand001) to remote host port 9000

Then can connect to the web server from the remote host

[jk24445@phlprlxjmp001 ~]$ netstat -lnt | grep 90

tcp 0 0 127.0.0.1:9000 0.0.0.0:\* LISTEN

[jk24445@phlprlxjmp001 ~]$ curl localhost:9000

Example 2:

[jk24445@phlprlxjmp001 ~]$ ssh -R 9000:sand001:80 localhost

Does the reverse of the example 1: Forward the webserver port which is now remote back to local

[jk24445@phlprlxjmp001 ~]$ netstat -ltn | grep 900

tcp 0 0 127.0.0.1:9000 0.0.0.0:\* LISTEN

[jk24445@phlprlxjmp001 ~]$ curl localhost:9000

Example 3:

ssh -R 2222:jimmy.servebeer.com:22 localhost

ssh ec2-user@localhost -p 2222

Tunnel thru a firewall

**Tidbits**

Can add -f and -N to back ground it, but will need to find the proc in ps and kill it

jk24445@phlprlxjmp001 ~]$ ssh -f -R 9000:phlutlxsand001:80 localhost -N