

A Tale From /dev/console: ssh Forward Port Forwarding

I am at my in-laws house mostly recovering from my father's 85th birthday party. I have a morning free while the rest of the family is deciding how to spend the day. (I am, I guess, technically on holiday.) I have used ssh port forwarding to get to a protocol on the other side of a firewall and I have always wanted to make sure I understood how ssh port forwarding works. So I did a little investigation.

Generally, ssh port forwarding allows you to access any remote port over a ssh connection by connecting to a local port. So, I spent a pleasant morning looking into ssh port forwarding.

There are 3 types of port forwarding: forward, reverse, and ftp. If you have the program sftp, I am not sure why you would want to use ftp port forwarding. I have just enough time to research forward-style port forwarding. Other mornings will be spent on the other 2 types.

I have a small 2-node network. I am running Fedora 10 without updates on 1 node, Comedy.class, 10.0.3.1/24, and Red Hat 5.3 non-Xen without updates on the other node, Sci-Fi.class, 10.0.3.2/24. I can ping in both directions and I am using /etc/hosts for name resolution. I am using Comedy as the client and Sci-Fi as the server.



Establishing the port forwarding tunnel

1. The command is `ssh -L 8000:sci-fi:25 root@sci-fi`
 1. The 8000 is the local port ssh will listen on. This is bound to 127.0.0.1. and will not allow connections from any other network address.
 2. sci-fi is the target host to transfer data via port 8000
 3. Port 25 is the destination port on sci-fi
 4. Normally I would not login via ssh but use a sleep command to hold the ssh tunnel open. I logged in here because I want to observe things on sci-fi.

Below is the actual command and some checking I did to see what was actually happening. The -v is to turn on the lowest level of debugging.

```
[arthur@Comedy Ssh]$ ssh -v -L 8000:sci-fi:25 root@sci-fi
OpenSSH_5.1p1, OpenSSL 0.9.8g 19 Oct 2007
debug1: Next authentication method: password
```

I removed a lot of debug1 statements here on authentication.

```
root@sci-fi's password:
debug1: Authentication succeeded (password).
```

The password was accepted and I have an authenticated account on the remote host, sci-fi.

```
debug1: Local connections to LOCALHOST:8000 forwarded to remote address
sci-fi:25
```

This is the start of setting up the port forwarding by the ssh command on comedy.

```
debug1: Local forwarding listening on :::1 port 8000.
debug1: channel 0: new [port listener]
```

This is the report that ssh on comedy is now listening on port 8000 for IPv6 traffic.

```
debug1: Local forwarding listening on 127.0.0.1 port 8000.
debug1: channel 1: new [port listener]
```

This is the report that ssh on comedy is now listening on port 8000 for Ipv4 traffic.

```
debug1: channel 2: new [client-session]
debug1: Entering interactive session.
debug1: Sending environment.
debug1: Sending env LANG = en_US.UTF-8
Last login: Tue Feb 17 16:58:46 2009
[root@sci-fi ~]#
```

I am logged on to sci-fi through an ssh connection.

I want to see the connection to comedy.

```
[root@sci-fi ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address      Foreign Address    State
tcp      0      0 Sci-fi.class:ssh Comedy.class:33309 ESTABLISHED
```

There is a connection and it is on port 33309 on comedy.

Now I want to find out something about the process that is managing the connection.

```
[root@sci-fi ~]# lsof -i :22
COMMAND PID USER  FD  TYPE DEVICE SIZE NODE NAME
sshd    6754 root  3u  IPv6 12932    TCP *:ssh (LISTEN)
sshd    8024 root  3u  IPv6 22778    TCP Sci-fi.class:ssh->Comedy.class:33309 (ESTABLISHED)
```

The connection is managed by the process with pid 8024 which is running a copy of sshd. I know this is the correct one as the line shows the connection is with comedy on port 33309.

Now to see has happened on comedy.

```
[arthur@Comedy Work]$ netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address      Foreign Address    State
tcp      0      0 Comedy.class:33309 Sci-fi.class:ssh  ESTABLISHED
```

Thankfully, I see the same connection on comedy as I did on sci-fi.

```
[arthur@Comedy Work]$ lsof -i :33309
COMMAND PID  USER  FD  TYPE DEVICE SIZE NODE NAME
OFF
ssh      4223 arthur 3u  IPv4 21707 0t0  TCP Comedy.class:33309->Sci-fi.class:ssh (ESTABLISHED)
```

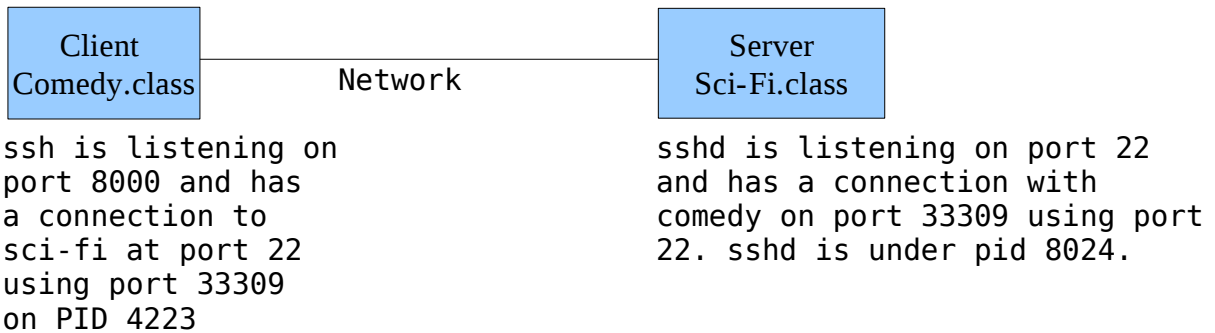
On comedy pid 4223 is running ssh which is managing the connection.

```
[arthur@Comedy Work]$ netstat -an | grep 8000
tcp      0      0 127.0.0.1:8000      0.0.0.0:*           LISTEN
tcp      0      0 :::8000             :::*               LISTEN
```

Some process is listening on port 8000 for both IPv4 and IPv6.

```
[arthur@Comedy Work]$ lsof -i :8000
COMMAND PID  USER  FD  TYPE DEVICE SIZE NODE NAME
OFF
ssh      4223 arthur 4u  IPv6 21717 0t0  TCP localhost6.localdomain6:8000 (LISTEN)
ssh      4223 arthur 5u  IPv4 21718 0t0  TCP localhost.localdomain:8000 (LISTEN)
```

It is pid 4223 running ssh that is listening on port 8000. This is the same process that has the encrypted connection to sci-fi port 22 with local port 33309.



Using the forwarding connection

1. Now on comedy execute telnet localhost port 8000 and see what happens.

```
[arthur@Comedy Telnet]$ telnet localhost 8000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 Sci-fi.class ESMTP Sendmail 8.13.8/8.13.8; Tue, 17 Feb 2009 18:17:14 -0800
helo comedy.class
250 Sci-fi.class Hello Sci-fi.class [10.0.3.2], pleased to meet you
```

The telnet command connected to port 8000 on local host. The ssh command forwarded the packets to port 25 of sci-fi through the connection on port 33309.

Notice that the return is from sci-fi.

I said helo comedy.class. The 250 return says the host is sci-fi.class. As we will see, this is correct as sendmail, the program listening on port 25, sees only the connection from sshd on sci-fi.

Checking the above statements.

```
[arthur@Comedy Work]$ netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost.localdomain:54930 localhost.localdomain:8000 ESTABLISHED
tcp        0      0 localhost.localdomain:8000 localhost.localdomain:54930 ESTABLISHED
tcp        0      0 Comedy.class:33309      Sci-fi.class:ssh       ESTABLISHED
```

The last line is the connection with sci-fi on port 33309. The first 2 lines are a connection between port 8000 and port 54930 on comedy, the localhost. As shown below port 8000 is associated with ssh and port 54930 is associated with telnet.

```
[arthur@Comedy Work]$ lsof -i :54930
COMMAND PID  USER  FD   TYPE DEVICE SIZE NODE NAME
ssh      4223 arthur 9u   IPv4 22016 0t0  TCP localhost.localdomain:8000->
        localhost.localdomain:54930 (ESTABLISHED)
telnet   4331 arthur 3u   IPv4 22015  0t0  TCP localhost.localdomain:54930->
        localhost.localdomain:8000 (ESTABLISHED)
```

The ssh pid is 4331, which is the pid for ssh with the connection to sci-fi on port 33309.

The telnet reads and writes from the window and sends the data to port 8000 which is read by ssh and sends it on to sshd on sci-fi with a request to forward it to port 25 on sci-fi.

Now to take a look at what is happening on sci-fi.

```
[root@sci-fi ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 Sci-fi.class:36347     Sci-fi.class:25        ESTABLISHED
tcp        0      0 Sci-fi.class:25        Sci-fi.class:36347     ESTABLISHED
tcp        0      0 Sci-fi.class:22        Comedy.class:33309     ESTABLISHED
```

The last line shows the connection with comedy. The first 2 lines show a connection between port 25 and port 36347 on sci-fi.

```
[root@sci-fi ~]# lsof -i :36347
COMMAND  PID USER FD  TYPE DEVICE SIZE NODE NAME
sshd     8024 root 10u IPv4 23101      TCP Sci-fi.class:36347->Sci-fi.class:25 (ESTABLISHED)
sendmail 8060 root  1u IPv4 23102      TCP Sci-fi.class:25->Sci-fi.class:36347 (ESTABLISHED)
sendmail 8060 root  4u IPv4 23102      TCP Sci-fi.class:25->Sci-fi.class:36347 (ESTABLISHED)
sendmail 8060 root  6u IPv4 23102      TCP Sci-fi.class:25->Sci-fi.class:36347 (ESTABLISHED)
```

This shows the connection between 25 and 36347 is between sshd and sendmail. This explains why the 250 message said connecting host was sci-fi. As far as sendmail is concerned it is.

Sequence review.

1. The command

```
ssh -L 8000:sci-fi:25 sherlock@sci-fi
```

1. Establishes a secure connection between the client, the host the ssh command was issued on and sci-fi, the server.
2. Sets up ssh on the client to listen on port 8000.
3. I would have added sleep 60 to automatically close the link between the client and the server after 60 seconds if there wasn't another connection going across the link.

2. The command

```
telnet localhost 8000
```

1. Sets up a connection to port 8000 on the localhost, the client machine.
2. Sends data from standard in to port 8000.
3. ssh is listening on port 8000 and sends the data to the sshd on the server with the request to forward this to port 25.
4. The sshd server sets up a connection to port 25, acting as the client to sendmail which is listening on port 25.
5. The sshd server returns the data across the connection to the ssh program saying it came from port 25 and is going to port 8000.
6. The ssh program on the client send the data back across the connection to telnet.
7. The telnet program displays the data to the window.

Some security thoughts.

1. The encrypted portion is only over the ssh/sshd connection.
2. You can get to the port forwarding only by being logged in on the client machine. This can be changed, but it changes the security on the connection.
3. There is only 1 ssh/sshd connection between the client and server.