6. SSH -R Remote Port Forward Listening on ens33

6.1 Overview

How could we modify our SSH command to allow another box (TARGET1) to connect to KALI while going through JUMPBOX1? Let's review the command and description used in the last section, but don't run the command.

```
ssh -p 22 nemo@192.168.1.220 -R 127.0.0.1:5000:127.0.0.1:5555
```

This sets up a remote port forward on JUMPBOX1's `127.0.0.1` interface on TCP 5000. You can verify this by typing `netstat -nat | egrep 5000` on JUMPBOX1. This instructs any traffic hitting TCP 5000 on the **127.0.0.1** interface of JUMPBOX1, to go through the SSH tunnel back to KALI, and after exiting the tunnel, connect to TCP 5555 on the `127.0.0.1` interface of KALI.

The first `127.0.0.1` is highlighted because this is what we are going to change. Instead of listening on `127.0.0.1` of JUMPBOX1, we are going to specify the external `ens33` interface of JUMPBOX1. So the new command becomes:

```
ssh -p 22 nemo@192.168.1.220 -R 192.168.1.220:5000:127.0.0.1:5555
```

If you didn't do it already, be sure to add "`GatewayPorts clientspecified`" to the `/etc/ssh/sshd_config` file and restart the SSH service for your JUMPBOX. Let's dive into an example to see it in action!

6.2 Netcat Chat

On KALI, start a Netcat listener on TCP 8888 on interface `127.0.0.1`.

```
nc -nv -l 127.0.0.1 8888
```

Ensure Netcat is listening by running and listening using this command on KALI:

```
netstat -natp | egrep 8888
```

From KALI, SSH into JUMPBOX1 and setup remote port forward to instruct all traffic hitting the `ens33` interface of JUMPBOX1 on TCP 8000 to connect to the `127.0.0.1` interface on TCP 8888 of KALI.

```
ssh -p 22 nemo@192.168.1.220 -R 192.168.1.220:8000:127.0.0.1:8888
```

For this demonstration, we are also going to initiate a vanilla SSH connection to TARGET1, in order to get a shell on the box.

```
ssh -p 22 nemo@192.168.1.230
```

So at this point, we have 2 SSH connections. The first is to setup our actual remote port forward. The second provides us a vanilla shell on TARGET1 that allows us to run Netcat. So let's try and connect to the Netcat listener on KALI from TARGET1 by going through JUMPBOX1. From the TARGET1 box, run this command to set up a simple chat program:

```
nc 192.168.1.220 8000

Hi KALI!
```

To summarize, we setup a remote port forward tunnel that routes any traffic hitting TCP 8000 on JUMPBOX1's *external* `ens33` interface, to go through the SSH tunnel, and connect to a Netcat listener running on TCP 8888 on KALI's `127.0.0.1` interface.

6.3 WWW Server to 127.0.0.1

With a reverse port forward, you can redirect TCP 80/443 traffic on a JUMPBOX back to a web server running on KALI. This is useful if you need to pull a file down from a web server. In this example, we'll be using a simple Python-based HTTP server on KALI.

```
python -m SimpleHTTPServer 8000
```

SSH into JUMPBOX1 as root because we are going to be listening on a privileged port (TCP 80). Then from TARGET1, use `wget` to retrieve a file from the "web server" on JUMPBOX1, when in reality, the file is being server from KALI.

```
ssh -p 22 root@192.168.1.220 -R 192.168.1.220:80:127.0.0.1:8000
```