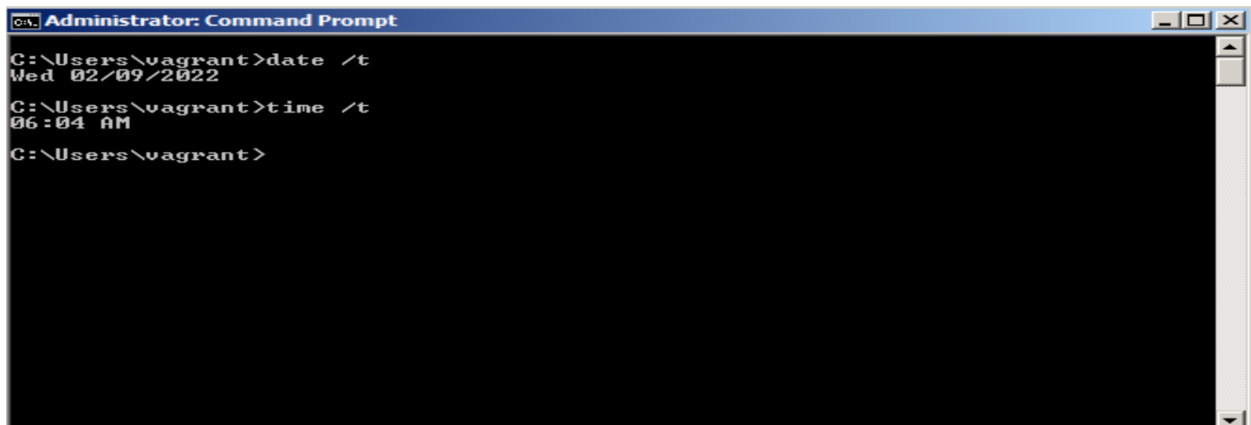


Lab 2: Target Virtual Machine

Part 1: Metasploitable 3:

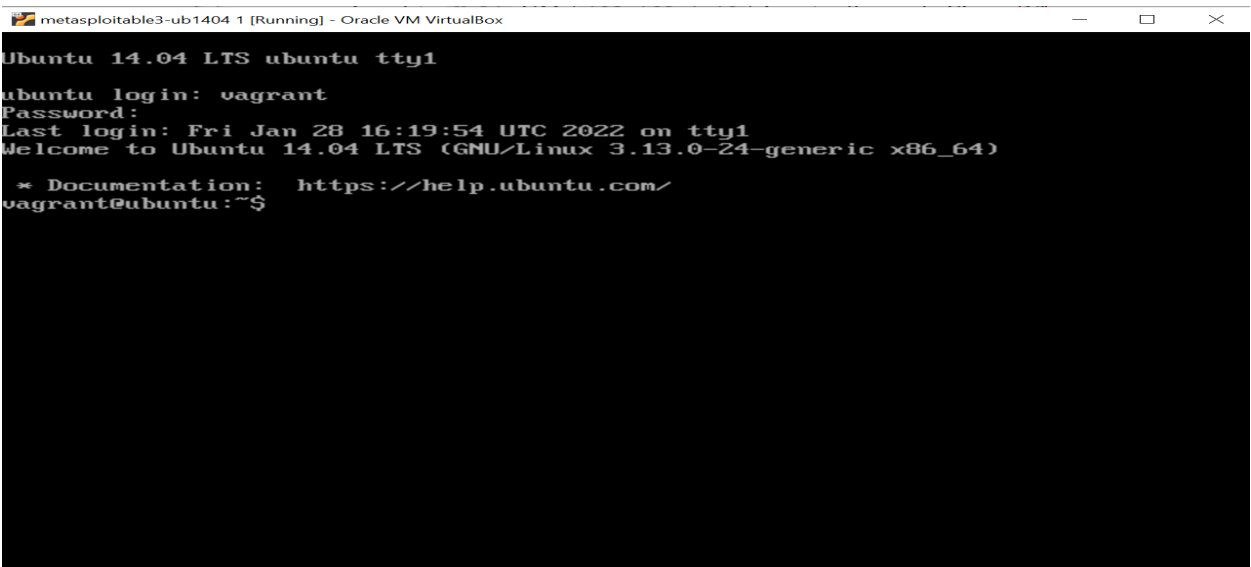
Virtualization is a powerful tool that allows ethical hackers to safely test different attacks, malware, and exploits on a multitude of different systems. The objective of this lab is to create two different Metasploitable 3 virtual machines (one running Windows 2008, the other running Ubuntu 14.04) for this purpose, and to establish a connection between them. The results are shown below.

Figure 1: Windows 2008 Metasploitable 3 VM

A screenshot of a Windows 2008 Metasploitable 3 virtual machine running in Oracle VM VirtualBox. The window title is "Administrator: Command Prompt". The command prompt shows the user 'vagrant' at the C: drive. The user has entered 'date /t' and 'time /t', which returned 'Wed 02/09/2022' and '06:04 AM' respectively. The prompt is currently at 'C:\Users\vagrant>'.

```
Administrator: Command Prompt
C:\Users\vagrant>date /t
Wed 02/09/2022
C:\Users\vagrant>time /t
06:04 AM
C:\Users\vagrant>
```

Figure 2: Ubuntu 14.04 Metasploitable VM

A screenshot of an Ubuntu 14.04 Metasploitable 3 virtual machine running in Oracle VM VirtualBox. The window title is "metasploitable3-ub1404 1 [Running] - Oracle VM VirtualBox". The terminal shows the Ubuntu login process for user 'vagrant'. It displays the login prompt, password prompt, last login time, and the Ubuntu version. The prompt is currently at 'vagrant@ubuntu:~\$'.

```
metasploitable3-ub1404 1 [Running] - Oracle VM VirtualBox
Ubuntu 14.04 LTS ubuntu tty1
ubuntu login: vagrant
Password:
Last login: Fri Jan 28 16:19:54 UTC 2022 on tty1
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
vagrant@ubuntu:~$
```

Part 2: Network Settings

VirtualBox supports many different methods of network virtualization, each with their own use cases. Host only networking is employed to connect all three virtual machines used in this lab, as shown below.

Figure 3: Network Adapter Configuration

Adapter DHCP Server

☐ Configure Adapter Automatically

☒ Configure Adapter Manually

IPv4 Address: 192.168.1.1

IPv4 Network Mask: 255.255.255.0

IPv6 Address: fe80::b9b5:6994:8c0c:ff3

IPv6 Prefix Length: 64

Apply Reset

Figure 4: DHCP Server Configuration

Adapter DHCP Server

☒ Enable Server

Server Address: 192.168.1.2

Server Mask: 255.255.255.0

Lower Address Bound: 192.168.1.3

Upper Address Bound: 192.168.1.254

Apply Reset

Part 3: Configure Kali to Use a Static IP Address

In order to communicate with the two Metasploitable 3 virtual machines, the Kali virtual machine must manually be assigned a static IP address, subnet mask, default gateway, and DNS server. Thankfully, these changes were made and verified quite easily with Kali's built in networking GUI and the command line.

Figure 5: Manually Editing the Network

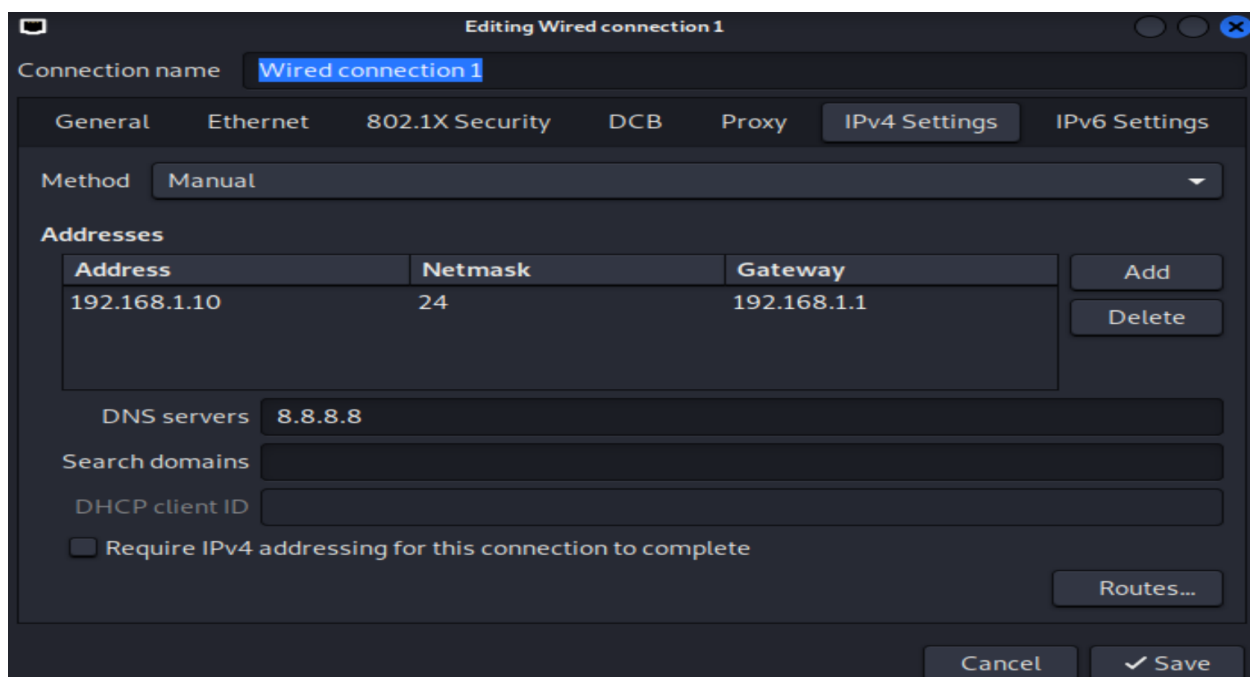
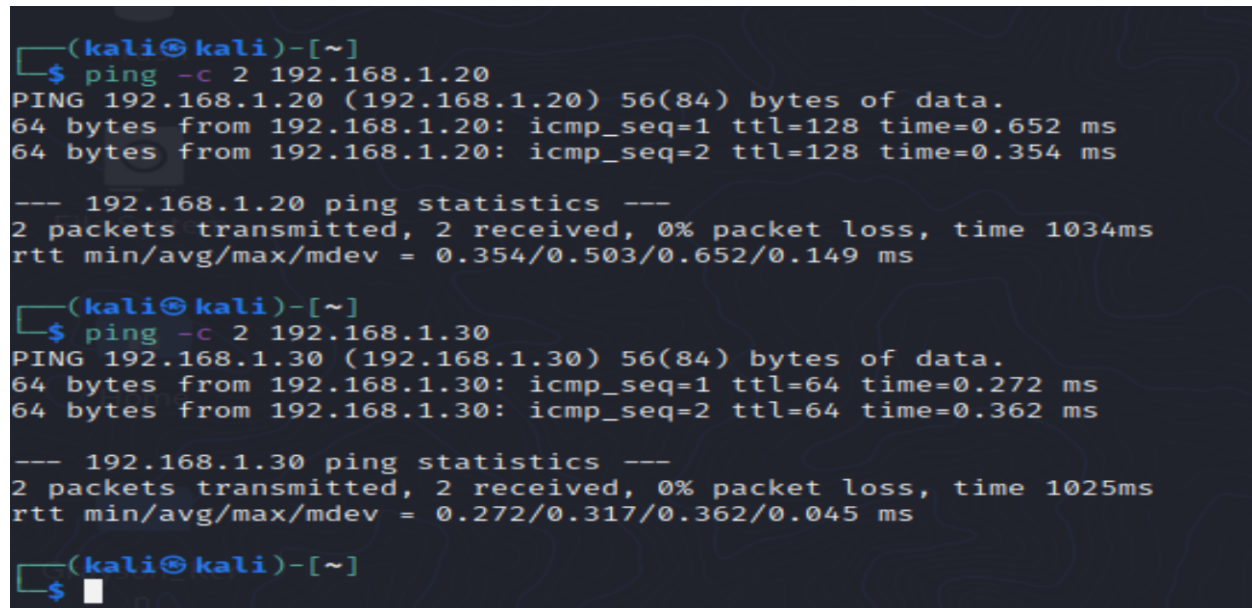


Figure 6: Verification of Network Changes

```
(kali@kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:50:4c:14 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.10/24 brd 192.168.1.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe50:4c14/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
(kali@kali)-[~]
$
```

The two Metasploitable guest machines are now accessible to Kali over the host-only network, as demonstrated below using the ping command.

Figure 7: Pinging Metasploitable Guest Machines



```
(kali㉿kali)-[~]
$ ping -c 2 192.168.1.20
PING 192.168.1.20 (192.168.1.20) 56(84) bytes of data.
64 bytes from 192.168.1.20: icmp_seq=1 ttl=128 time=0.652 ms
64 bytes from 192.168.1.20: icmp_seq=2 ttl=128 time=0.354 ms

--- 192.168.1.20 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1034ms
rtt min/avg/max/mdev = 0.354/0.503/0.652/0.149 ms

(kali㉿kali)-[~]
$ ping -c 2 192.168.1.30
PING 192.168.1.30 (192.168.1.30) 56(84) bytes of data.
64 bytes from 192.168.1.30: icmp_seq=1 ttl=64 time=0.272 ms
64 bytes from 192.168.1.30: icmp_seq=2 ttl=64 time=0.362 ms

--- 192.168.1.30 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1025ms
rtt min/avg/max/mdev = 0.272/0.317/0.362/0.045 ms

(kali㉿kali)-[~]
$
```

Part 4: Shell Scripts

Shell scripts are yet another tool in the ethical hacker's toolkit, and are widely used throughout the industry to automate various tasks. One such example is the problem of scanning for active hosts on a network, which can become quite tedious as the size of the network grows. Thankfully, this process can be easily automated using a few common UNIX commands embedded in a shell script. The source code of such a script is shown below in figure 8, and the results of running it are shown in figures 9 and 10. When the two metasploitable virtual machines are running, the script picks up each of their IP addresses, and when they are not, only the DHCP server and the Kali VM itself are detected.

Figure 8: pingScan.sh

```
#!/usr/bin/bash
if [ -z "$1" ]
then
    echo "Usage: ./pingScan [network]"
    echo "Ex: ./pingScan.sh 192.168.0"
else
    for i in {1..254}
    do
        (ping -c 1 $1.$i | grep "bytes from" | cut -d " " -f 4 | tr -d ":" &)
    done
fi
```

Figure 9: Detection of Metasploitable VMs

```
(kali㉿kali)-[~/Desktop/Grayson_Kern/Lab/Lab 2]
$ ./pingScan.sh 192.168.1
192.168.1.2
192.168.1.10
192.168.1.20
192.168.1.30
```

Figure 10: Detection of DHCP Server

```
(kali㉿kali)-[~/Desktop/Grayson_Kern/Lab/Lab 2]
$ ./pingScan.sh 192.168.1
192.168.1.2
192.168.1.10
```