

SOC Lab: To Investigate and Simulate Attacks

Lab Setup:

1. Network Topology



2. Installation Steps

- I downloaded and installed VirtualBox to create a virtual environment for my attack simulation, within the VirtualBox I set the operating systems using their ISO and image file.

Kali Linux – It's Linux distribution designed for penetration testing. In this setup it will be used to simulate the attacker machine.

Windows 10 – It's A widely-used operating system. In the setup it will serve as the target machine to mimic a real-world user environment.

Metasploitable – It's a tool designed for testing and practicing exploitation techniques.

Below are the officially download links for each component mentioned above:

- VirtualBox: <https://www.virtualbox.org/wiki/Downloads>
- Kali Linux and ISO: <https://www.kali.org/get-kali/>
- Windows 10 and ISO: <https://www.microsoft.com/software-download/windows10>
- Metasploitable 2 VM Image: <https://sourceforge.net/projects/metasploitable/>

3. VM Configuration

After installing the virtual machines, I configured the network settings for each VM by enabling and adjusting the network adapters. The configuration details are illustrated in the image below.

VM Name	Adapter 1	Adapter 2	Purpose
Kali Linux	NAT Network	Host-Only Adapter	Internet access + communicate with Windows
Windows 10	Host-Only Adapter	Bridged Adapter	Communicate with Kali + Internet access
Metasploitable	NAT Network	—	Communicate with Kali (via NAT network)

For my vulnerable machine (Metasploitable), I needed to ensure that Kali Linux and Metasploitable could communication effectively within the virtual environment. And to achieve this, I created a custom NAT network using the VirtualBox Network Manager.

I named the network "meta-lab", set the ipv4 prefix to 10.0.2.0/24 and enabled DHCP to allow automatic Ip addressing. Next, I attached the "meta-lab" network to Adapter 1 of

both Kali Linux and Metasploitable, enabling a communication between the two machines. The final configuration is shown in the image below.

VM Name	Adapter	Attached To	Network Name	IP Range	Promiscuous Mode	DHCP
Kali Linux	Adapter 1	NAT Network	meta-lab	10.0.2.0/24	Allow VMs	<input checked="" type="checkbox"/>
	Adapter 2	Host-Only Adapter	vboxnet	192.168.x.x	—	—
Metasploitable	Adapter 1	NAT Network	meta-lab	10.0.2.0/24	Allow VMs	<input checked="" type="checkbox"/>
Windows 10	Adapter 1	Host-Only Adapter	vboxnet	192.168.x.x	—	—
	Adapter 2	Bridged Adapter	—	192.168.1.x	—	—

The image below shows the network interfaces of each virtual machine, including their adapter settings and assigned networks.

VM Name	Adapter	Attached To	IP Address	Interface
Kali Linux	Adapter 1	NAT Network	10.0.2.5	eth0
	Adapter 2	Host-Only Adapter	192.168.56.102	eth1
Metasploitable	Adapter 1	NAT Network	10.0.2.4	eth0
Windows 10	Adapter 1	Host-Only Adapter	192.168.56.103	eth0
	Adapter 2	Bridged Adapter	192.168.1.5	eth2

The next step was to verify network connectivity between the virtual machines. I ensured that:

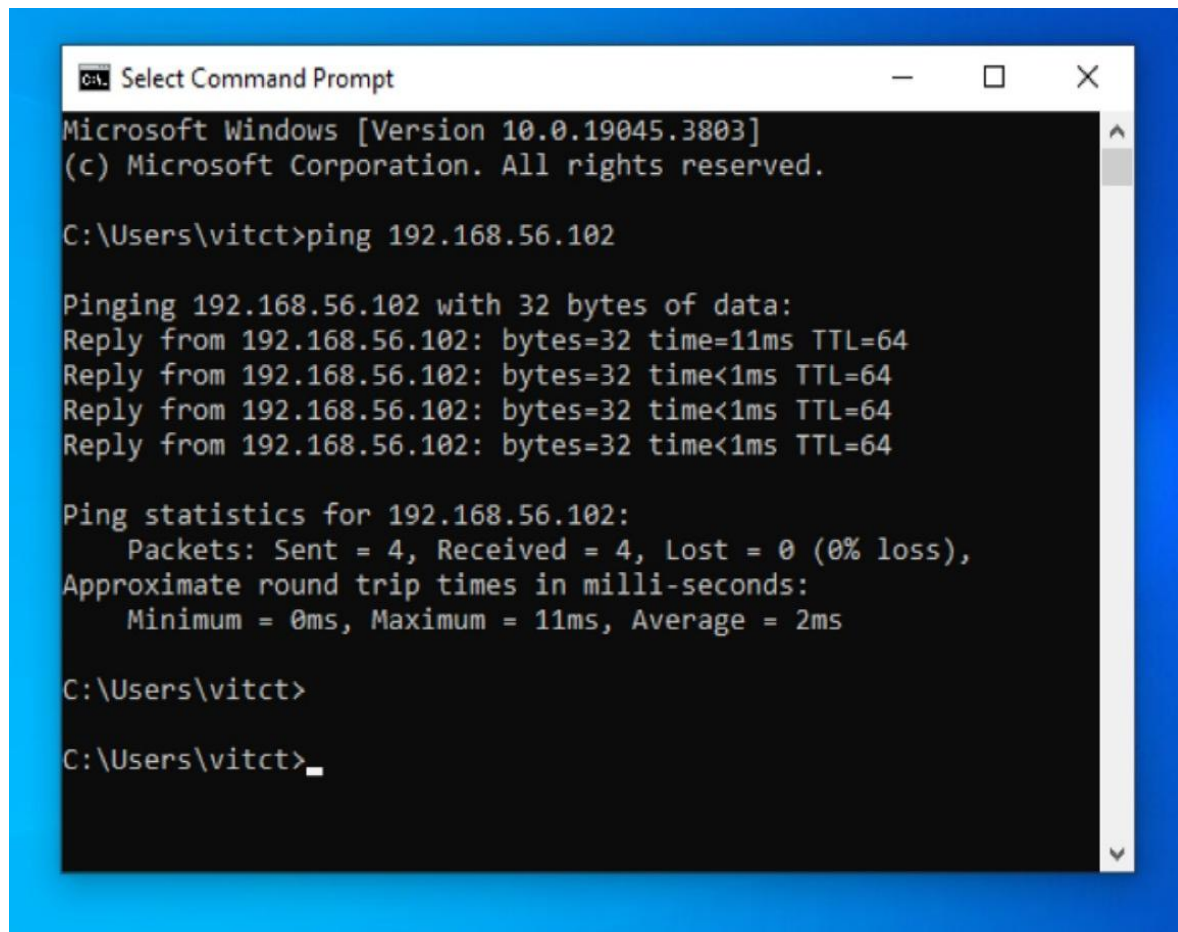
- Kali Linux can ping Metasploitable, and Metasploitable can also ping Kali Linux successfully.

```
notroot@kali: ~  
File Actions Edit View Help  
(notroot@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=8.52 ms  
64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=0.511 ms  
64 bytes from 10.0.2.4: icmp_seq=3 ttl=64 time=0.660 ms  
64 bytes from 10.0.2.4: icmp_seq=4 ttl=64 time=0.492 ms  
64 bytes from 10.0.2.4: icmp_seq=5 ttl=64 time=0.378 ms  
^C  
--- 10.0.2.4 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4043ms  
rtt min/avg/max/mdev = 0.378/2.111/8.517/3.203 ms  
(notroot@kali)-[~]  
$
```

```
msfadmin@metasploitable:~$  
msfadmin@metasploitable:~$  
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=12.6 ms  
64 bytes from 10.0.2.5: icmp_seq=2 ttl=64 time=27.3 ms  
64 bytes from 10.0.2.5: icmp_seq=3 ttl=64 time=0.507 ms  
64 bytes from 10.0.2.5: icmp_seq=4 ttl=64 time=0.309 ms  
64 bytes from 10.0.2.5: icmp_seq=5 ttl=64 time=0.376 ms  
64 bytes from 10.0.2.5: icmp_seq=6 ttl=64 time=0.316 ms  
64 bytes from 10.0.2.5: icmp_seq=7 ttl=64 time=10.4 ms  
64 bytes from 10.0.2.5: icmp_seq=8 ttl=64 time=0.277 ms  
64 bytes from 10.0.2.5: icmp_seq=9 ttl=64 time=0.378 ms  
--- 10.0.2.5 ping statistics ---  
9 packets transmitted, 9 received, 0% packet loss, time 7997ms  
rtt min/avg/max/mdev = 0.277/5.843/27.351/8.884 ms  
msfadmin@metasploitable:~$
```

- Kali Linux can ping Windows 10, and Windows 10 can also ping Kali Linux successfully.

```
notroot@kali: ~  
File Actions Edit View Help  
(notroot@kali)-[~]  
$ ping 192.168.56.103  
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.  
64 bytes from 192.168.56.103: icmp_seq=1 ttl=128 time=1.25 ms  
64 bytes from 192.168.56.103: icmp_seq=2 ttl=128 time=56.5 ms  
64 bytes from 192.168.56.103: icmp_seq=3 ttl=128 time=6.48 ms  
64 bytes from 192.168.56.103: icmp_seq=4 ttl=128 time=3.22 ms  
^C  
--- 192.168.56.103 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3144ms  
rtt min/avg/max/mdev = 1.249/16.862/56.500/22.960 ms  
(notroot@kali)-[~]  
$
```



```
C:\Users\vitct>ping 192.168.56.102

Pinging 192.168.56.102 with 32 bytes of data:
Reply from 192.168.56.102: bytes=32 time=11ms TTL=64
Reply from 192.168.56.102: bytes=32 time<1ms TTL=64
Reply from 192.168.56.102: bytes=32 time<1ms TTL=64
Reply from 192.168.56.102: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.56.102:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 11ms, Average = 2ms

C:\Users\vitct>
C:\Users\vitct>_
```

The screenshots are taken from VirtualBox, to show the successful communication between the virtual machines.