Project Name: **Creating a Secure Hacking Lab**

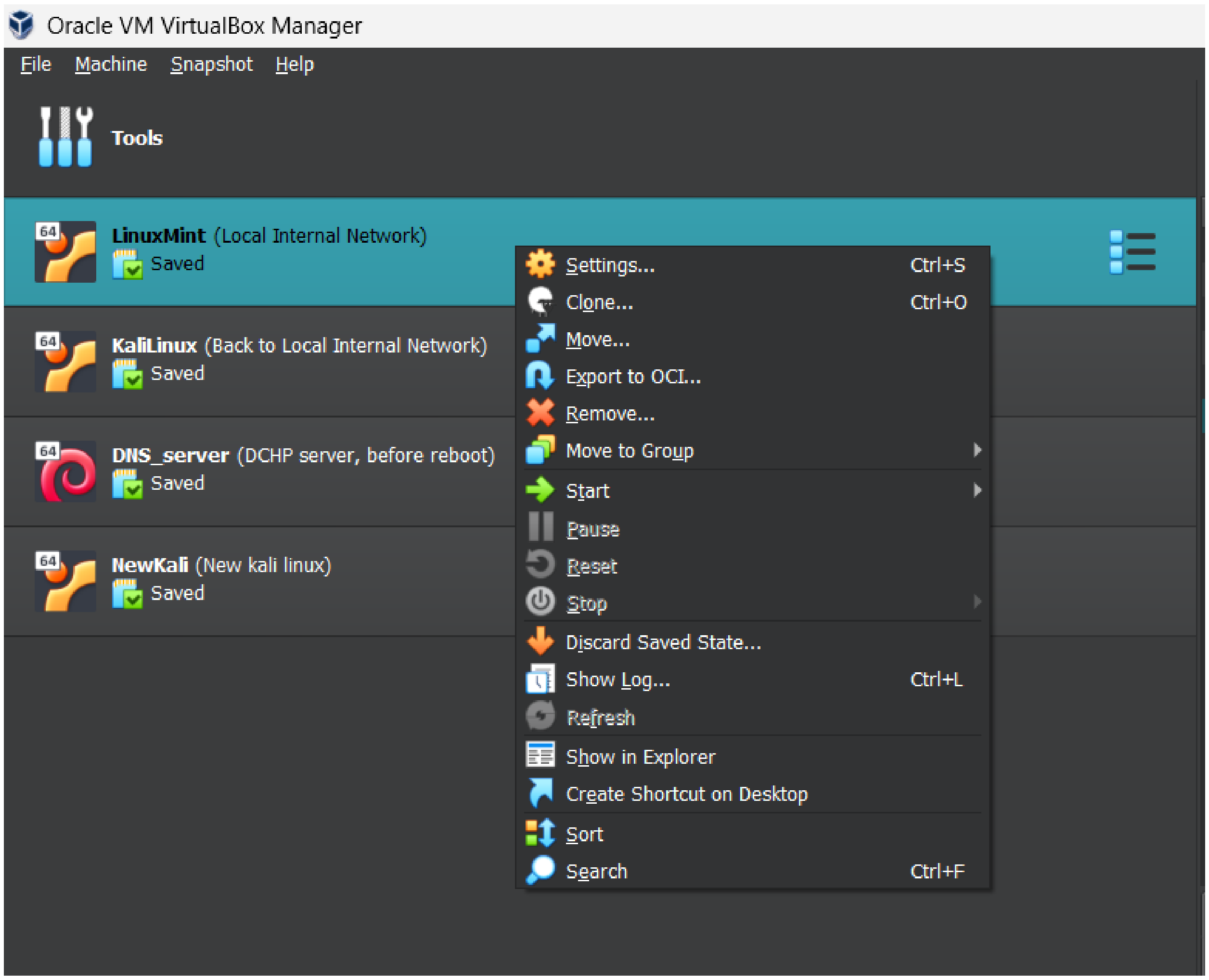
Date Completed: **August 5, 2023**

Created and performed by: **Jason Patrick Salerno**

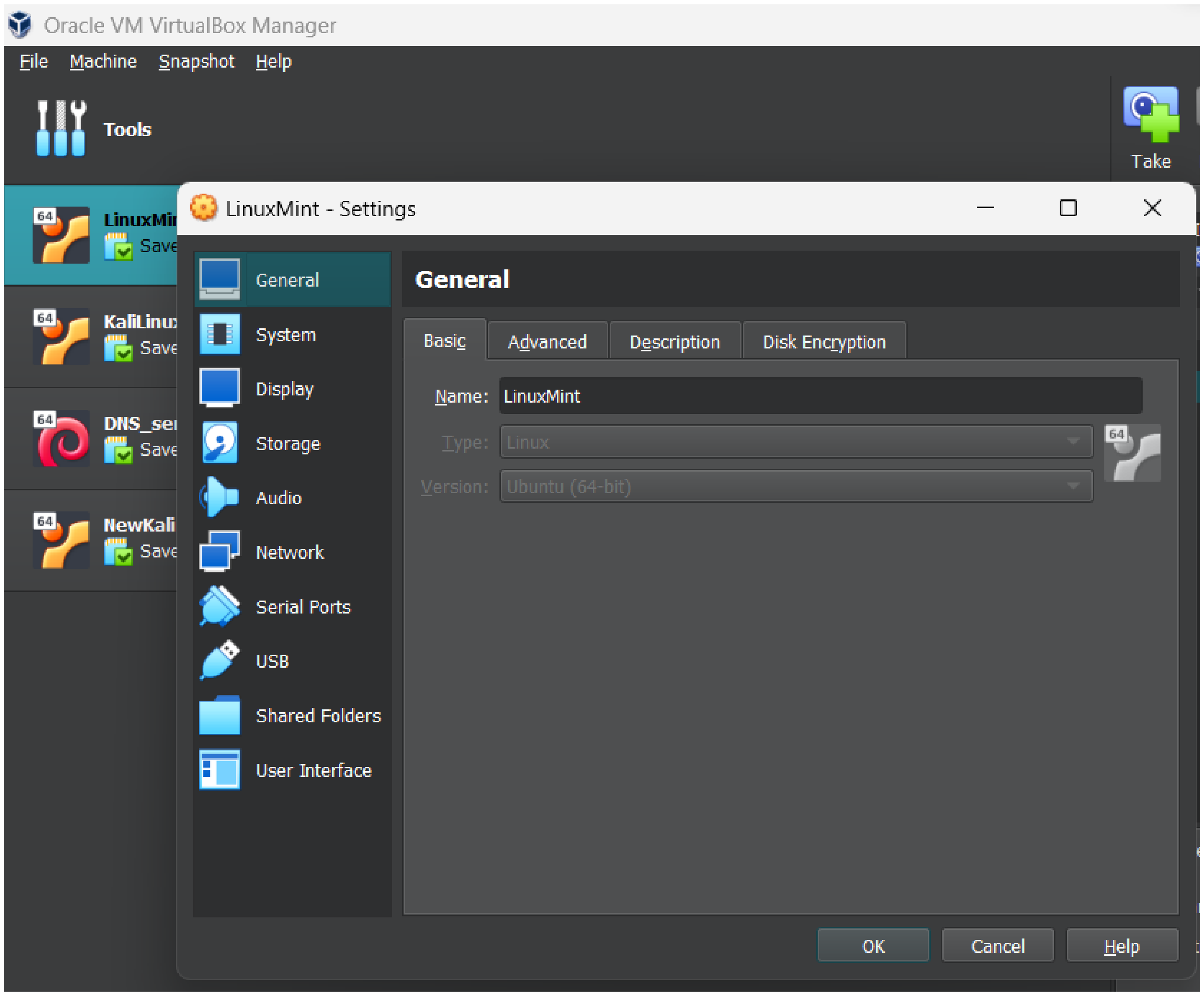
Purpose: **To Help Cyber Students Create Their Own Secure Learning Lab**

Part 1: Setting Up an Internal Network

1. I open **Oracle VM VirtualBox,** then right click on **LinuxMint VM** > **Settings.**



2. After the Settings have now opened.

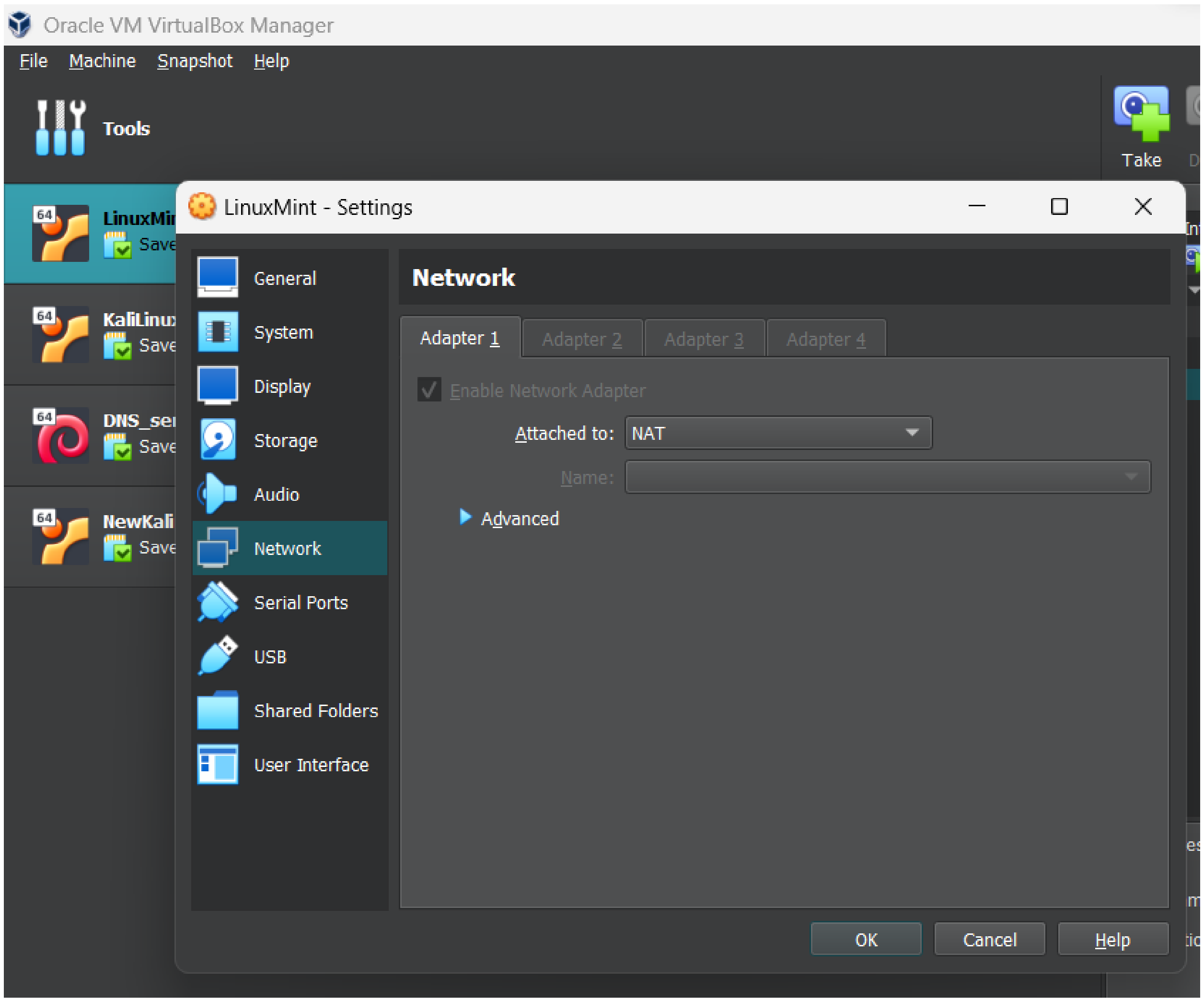


Name: LinuxMint

Type: Linux

Version: Ubuntu (64-bit)

3. Next, I navigate to the **Network** tab, where we can see the VM's network settings.


Oracle VM VirtualBox Manager

File Machine Snapshot Help

LinuxMint - Settings

General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

Enable Network Adapter: ✓

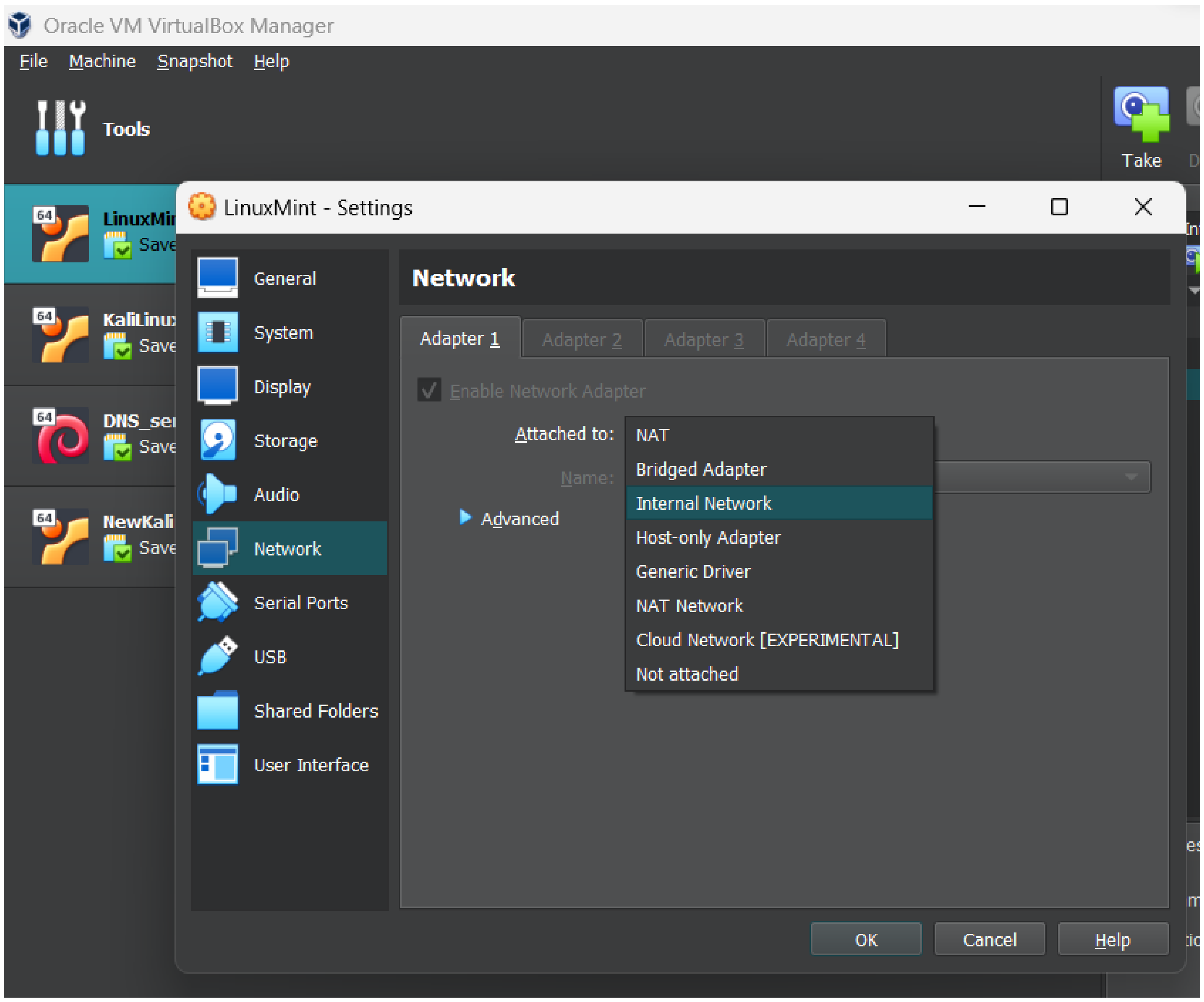
Attached to: NAT

Name:

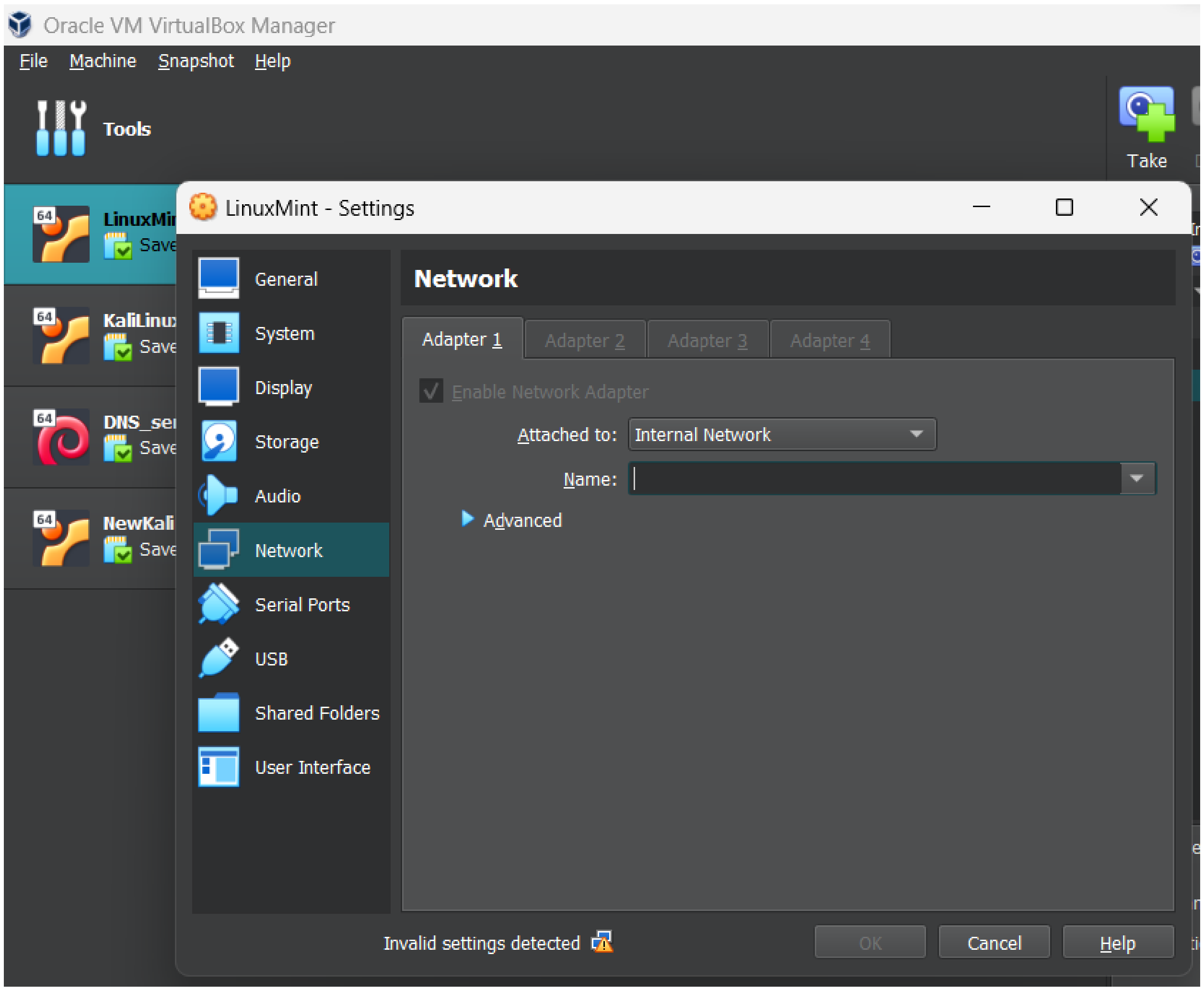
Advanced

OK Cancel Help

4. Then I click on **Attached to** then navigate to the Internal **Network**.



5. After selecting **Internal Network,** next in the **Name Box** you can type whatever name you want to call your internal network, so I name my internal network: **CyberLab.**


Oracle VM VirtualBox Manager

LinuxMint - Settings

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

[✓] Enable Network Adapter

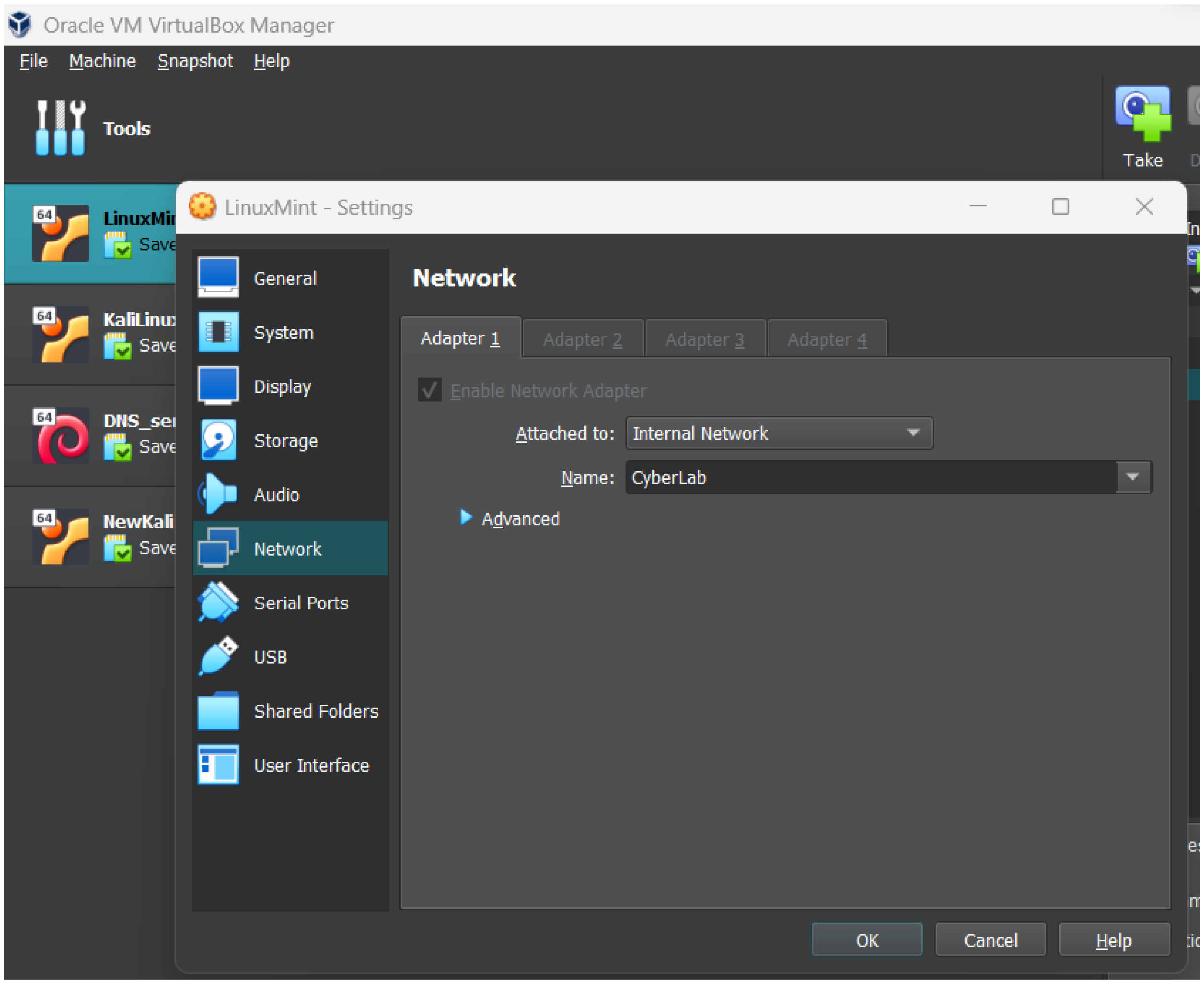
Attached to: Internal Network

Name:

Advanced

Invalid settings detected

6. So, I have named my internal network **CyberLab**.



Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

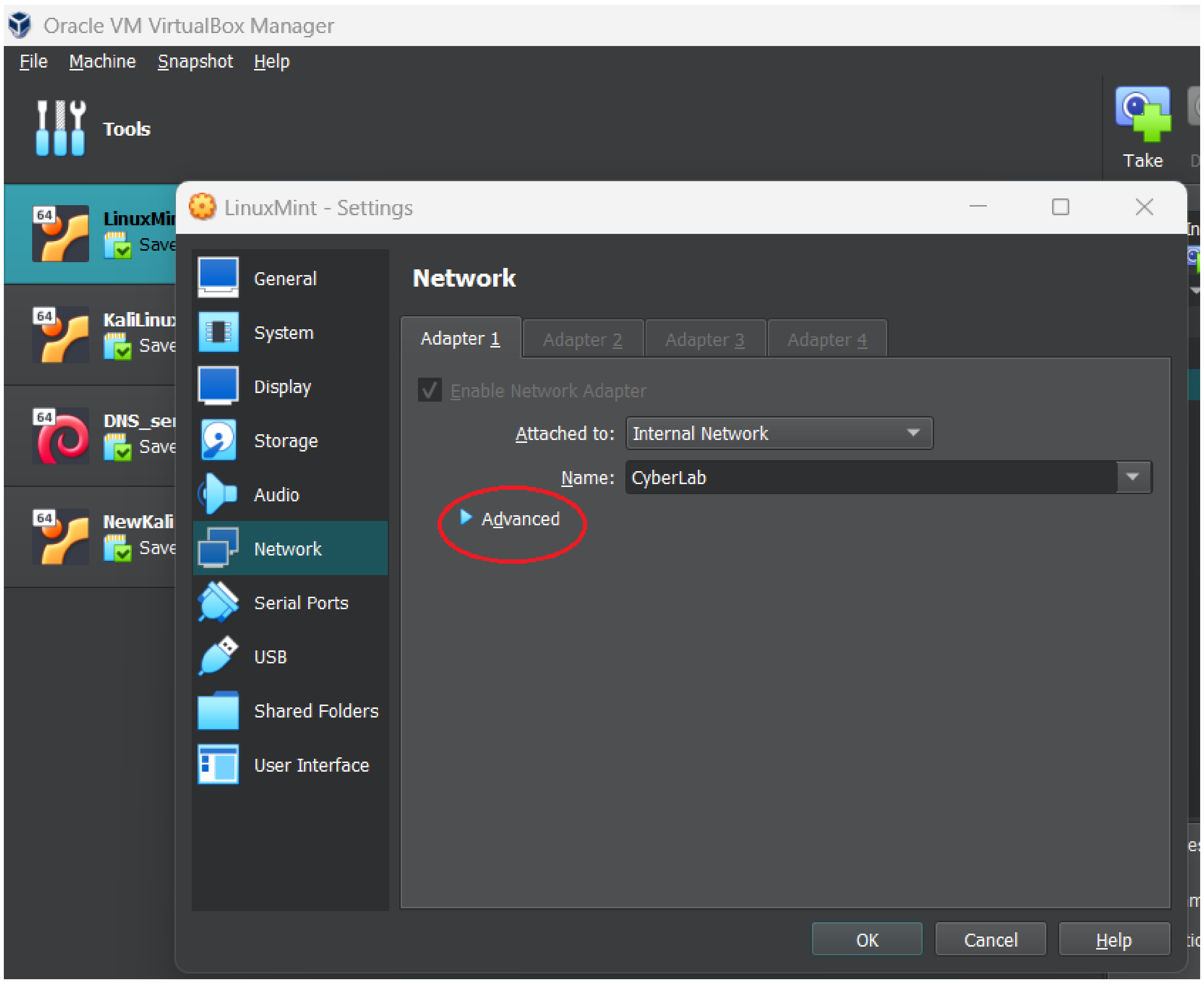
☑ Enable Network Adapter

Attached to: Internal Network

Name: CyberLab

Advanced

7. The next step is to click on **Advanced**.



Network

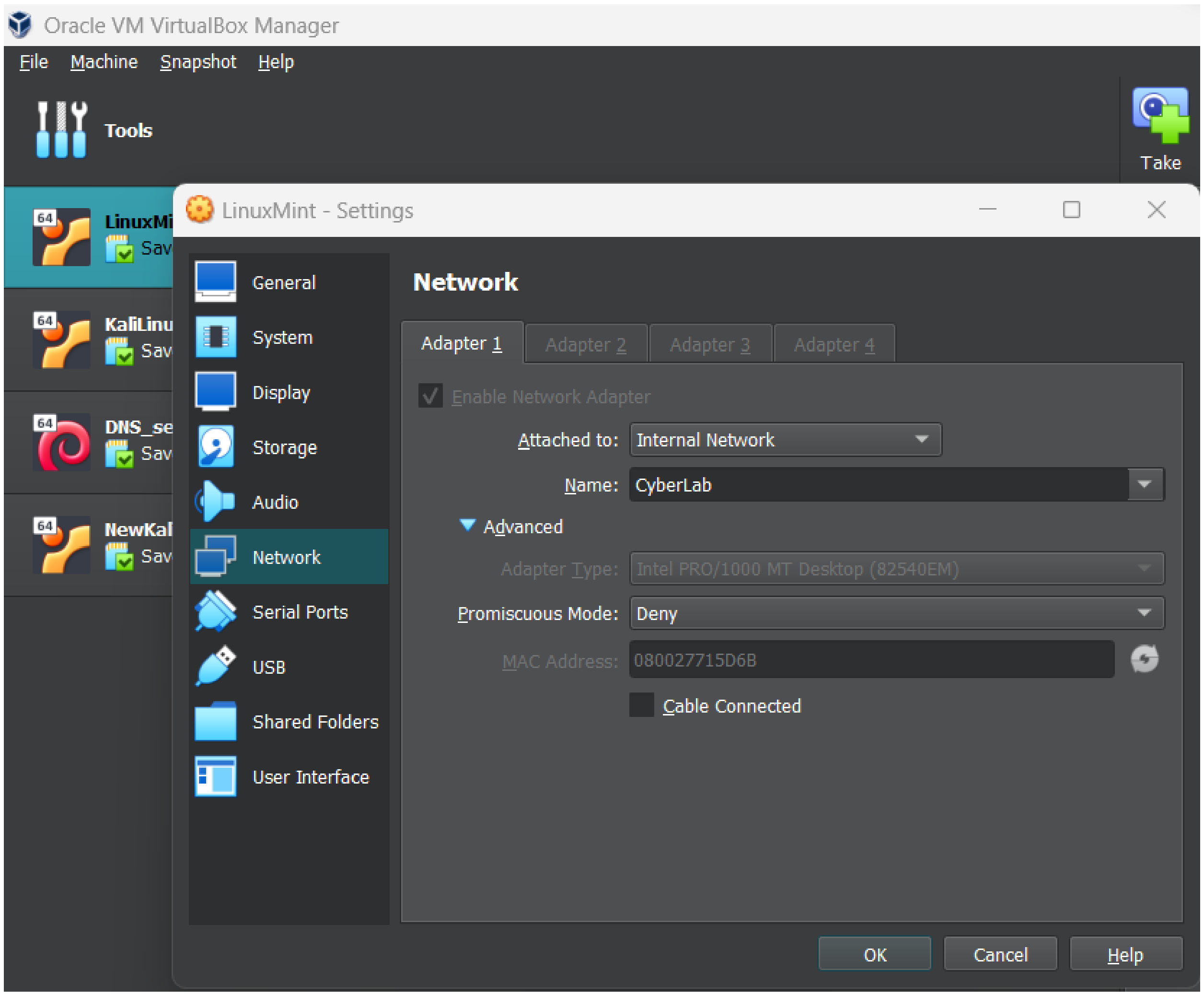
Enable Network Adapter

Attached to: Internal Network

Name: CyberLab

Advanced

8. Here it displays its drop-down options, which are **Promiscuous Mode**, and **Cable Connected**.



Enable Network Adapter

Attached to: Internal Network

Name: CyberLab

Advanced

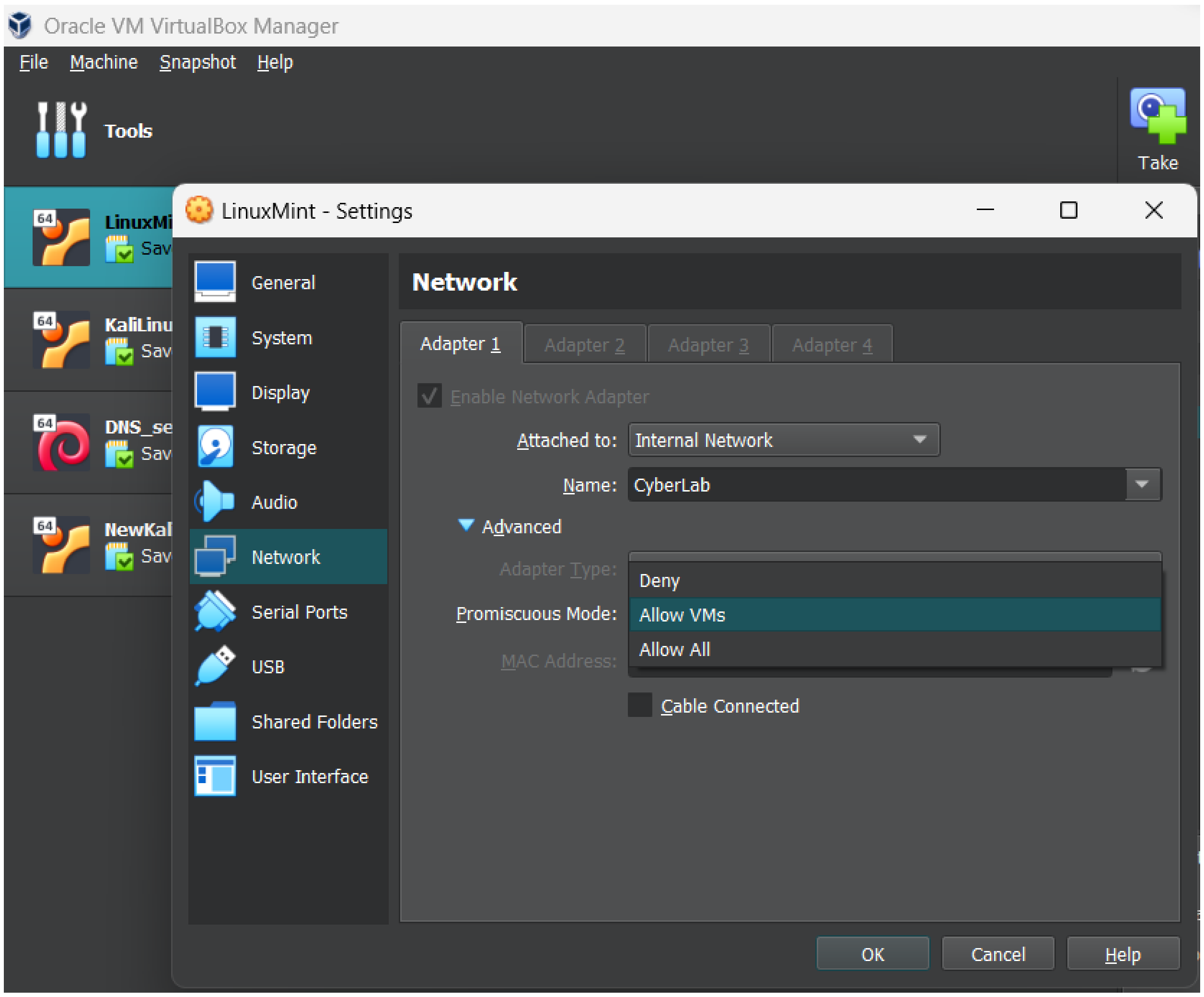
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Deny

MAC Address: 080027715D6B

Cable Connected

9. Next, I click on **Allow VMs**. The "Allow VMs" mode is typically used when you wish to promote communication between VMs operating on the same VirtualBox host but do not want VMs from different hosts on the same physical network to speak with one another.


Oracle VM VirtualBox Manager

File Machine Snapshot Help

LinuxMint - Settings

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

Enable Network Adapter

Attached to: Internal Network

Name: CyberLab

Advanced

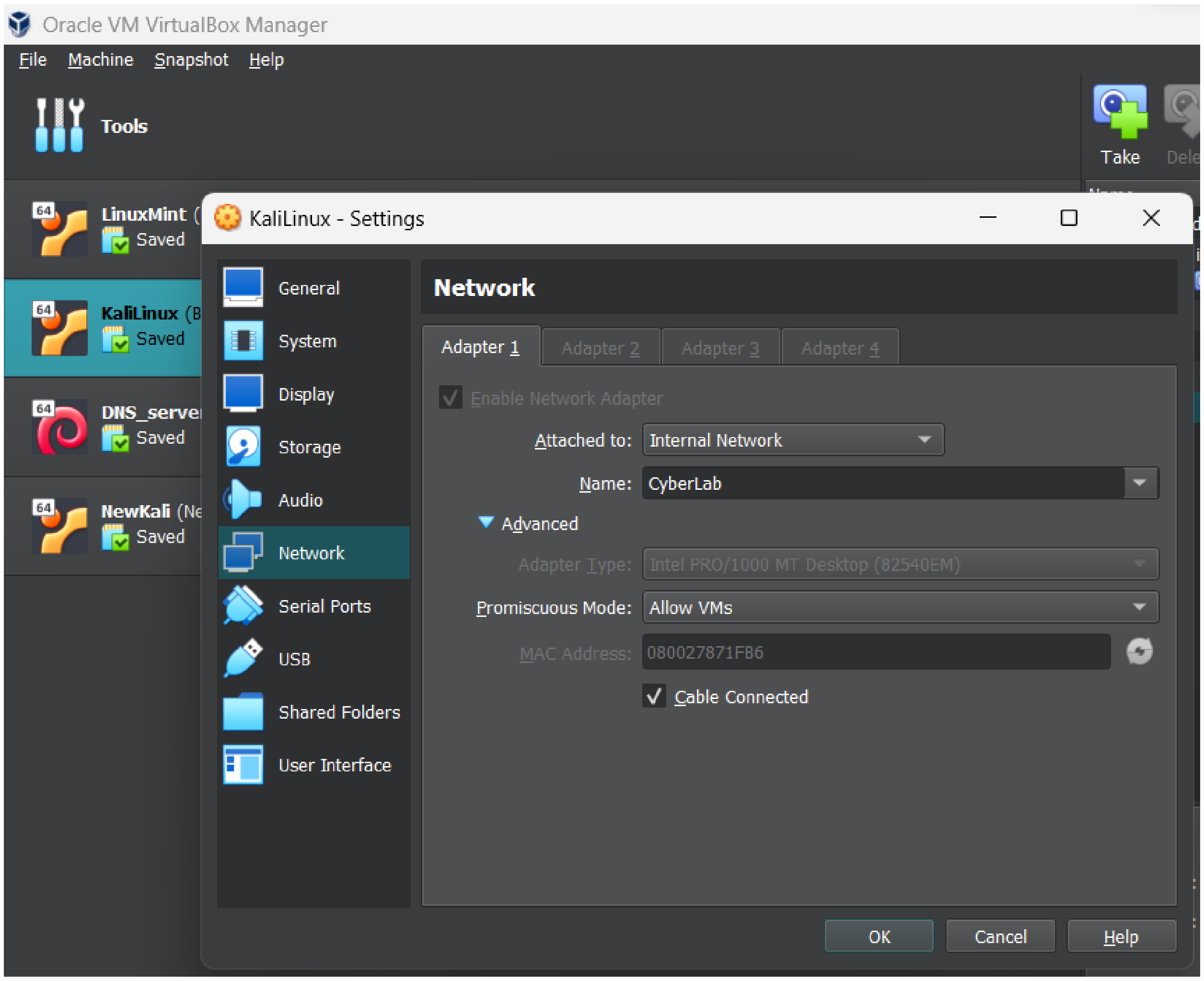
Adapter Type:

Promiscuous Mode: Deny / Allow VMs / Allow All

MAC Address:

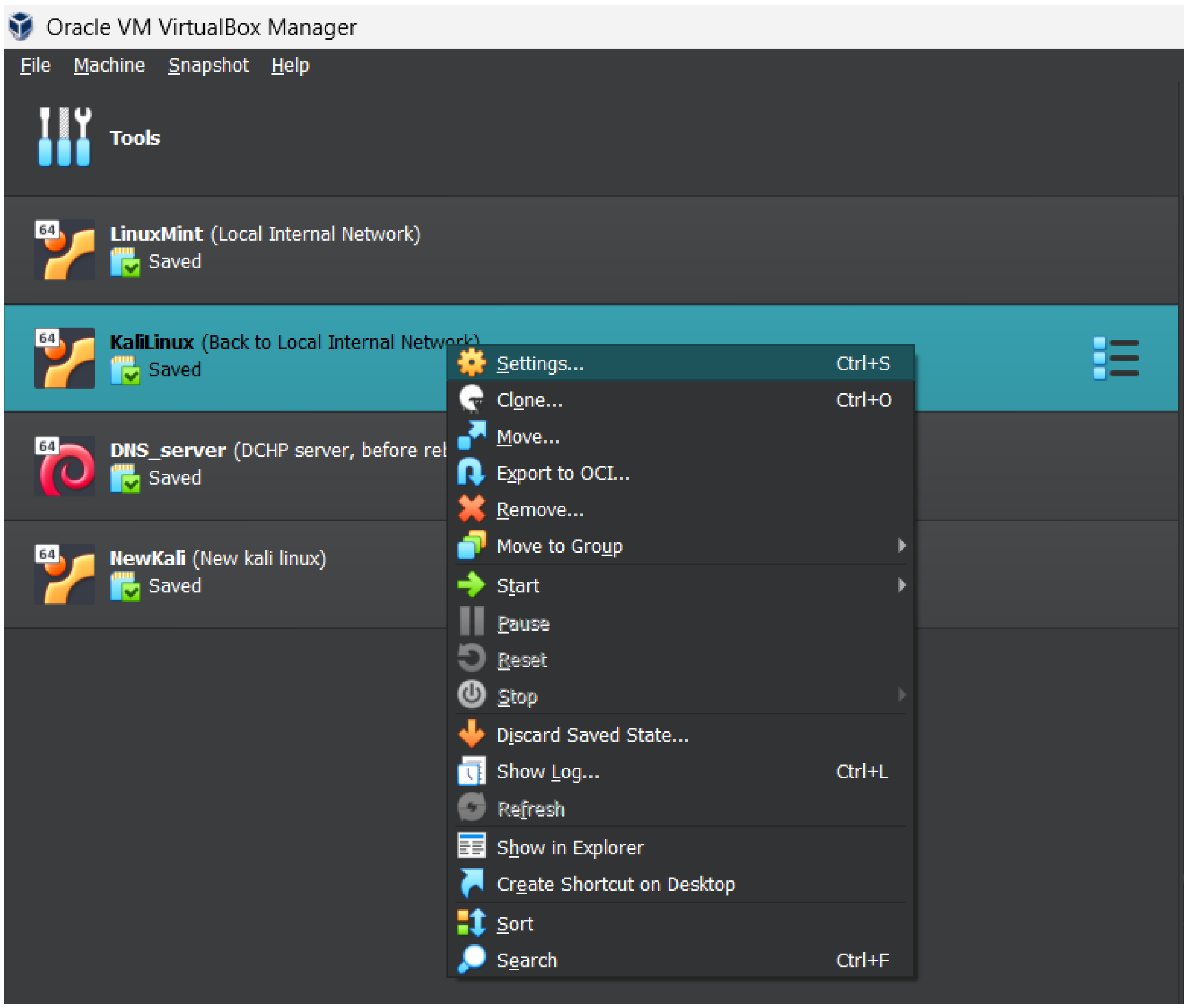
Cable Connected

10. After selecting **Allow Vms**, I click the checkbox. What does checking the checkbox do? well it regulates the virtual network adapter's network connection, emulating the behavior of an actual network cable being connected or unplugged. After the checkbox I click on **OK** at the bottom to save our changes.

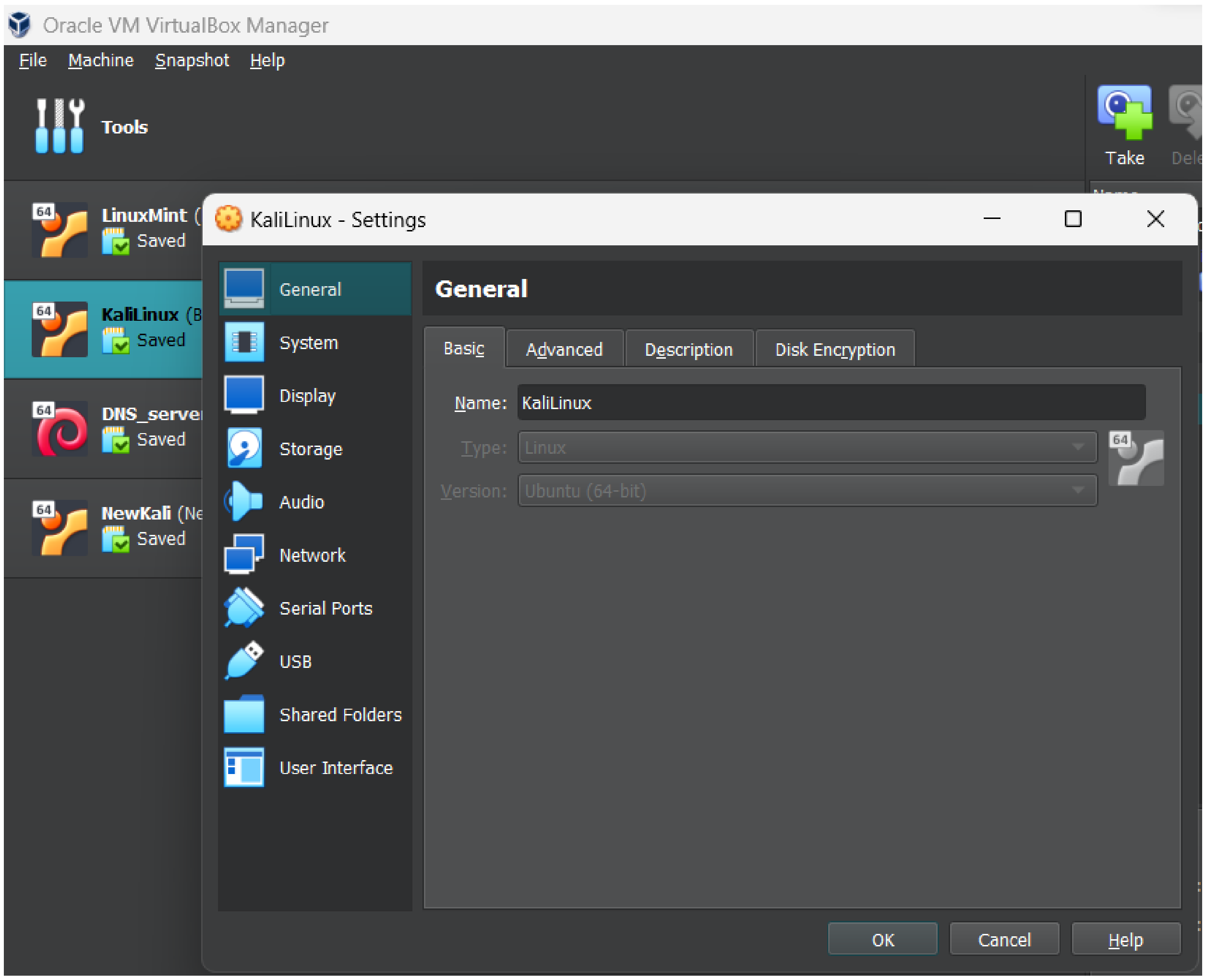


✓ Enable Network Adapter
Attached to: Internal Network
Name: CyberLab
Advanced
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)
Promiscuous Mode: Allow VMs
MAC Address: 080027871FB6
✓ Cable Connected

11. Next will do the same process for the Kali linux VM, so I right click then click settings.



12. Next I will do the same thing for my **kali linux VM**.



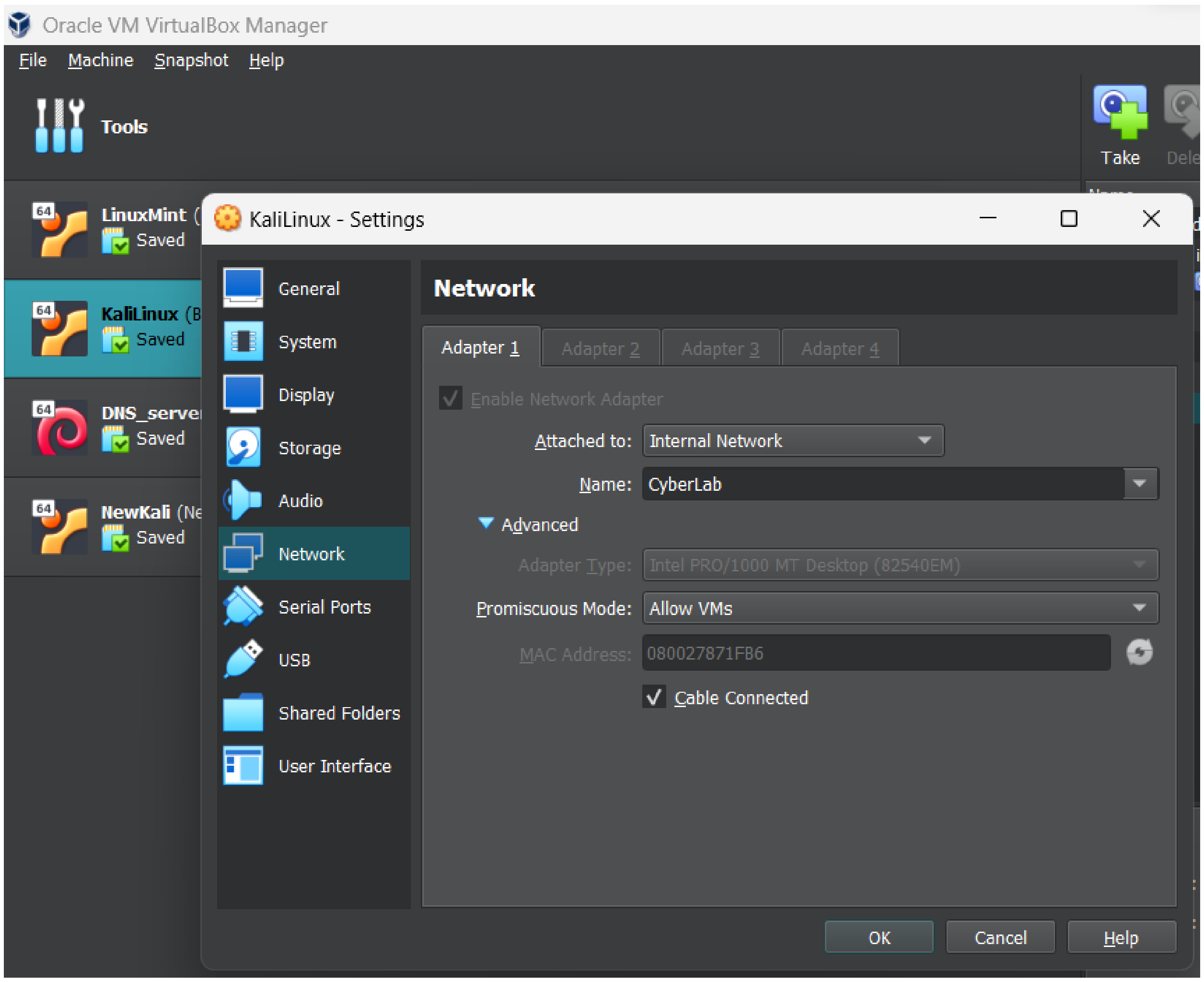
General

Name: KaliLinux

Type: Linux

Version: Ubuntu (64-bit)

13. So this is pretty much the same process as on the linux mint VM, so I click **settings > Network > Name:** I add the name of my internal network called **CyberLab**, then **Promiscuous Mode:** Select **Allow VMs** > click the **checkbox**.



Network

✓ Enable Network Adapter

Attached to: Internal Network

Name: CyberLab

Advanced

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

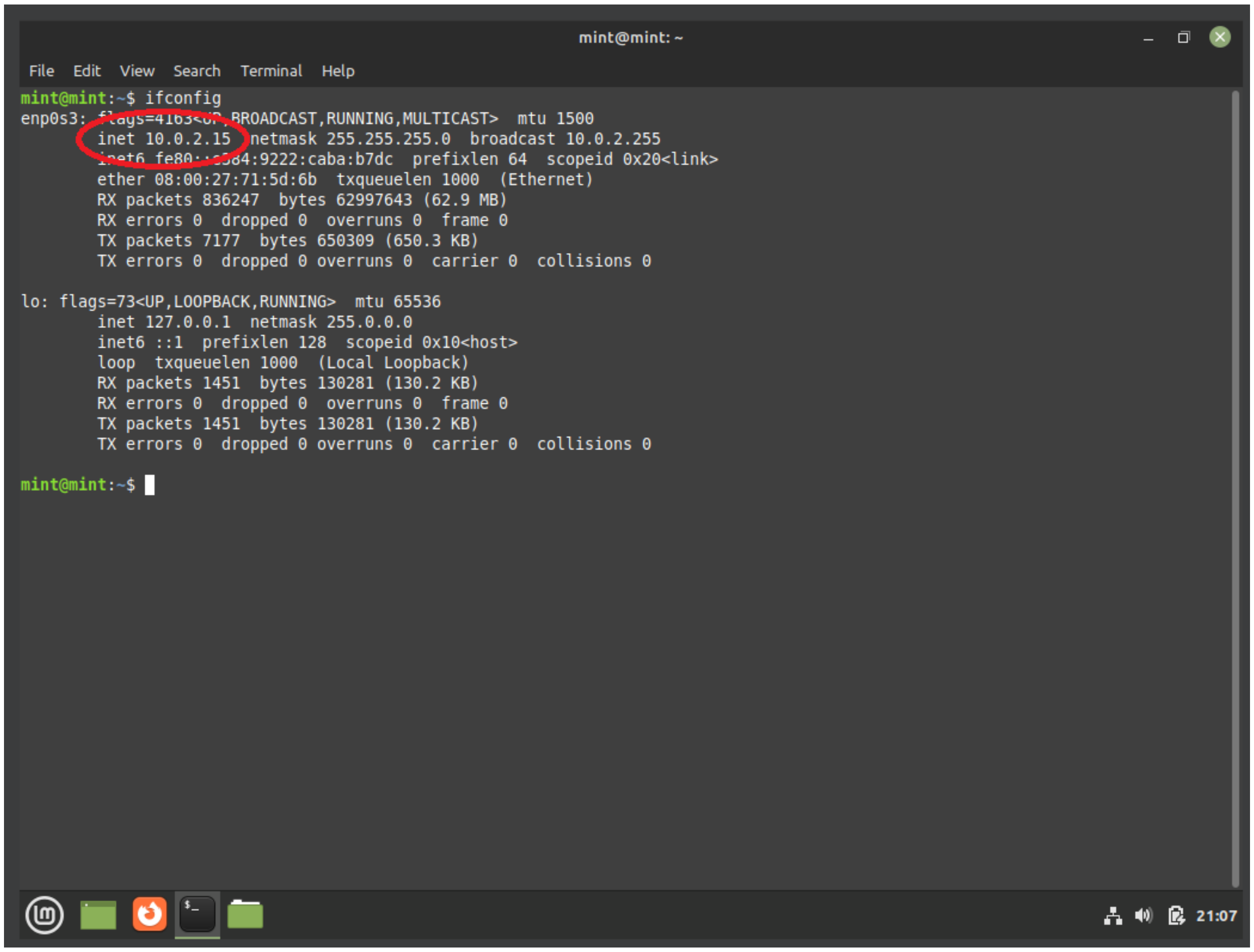
Promiscuous Mode: Allow VMs

MAC Address: 080027871FB6

✓ Cable Connected

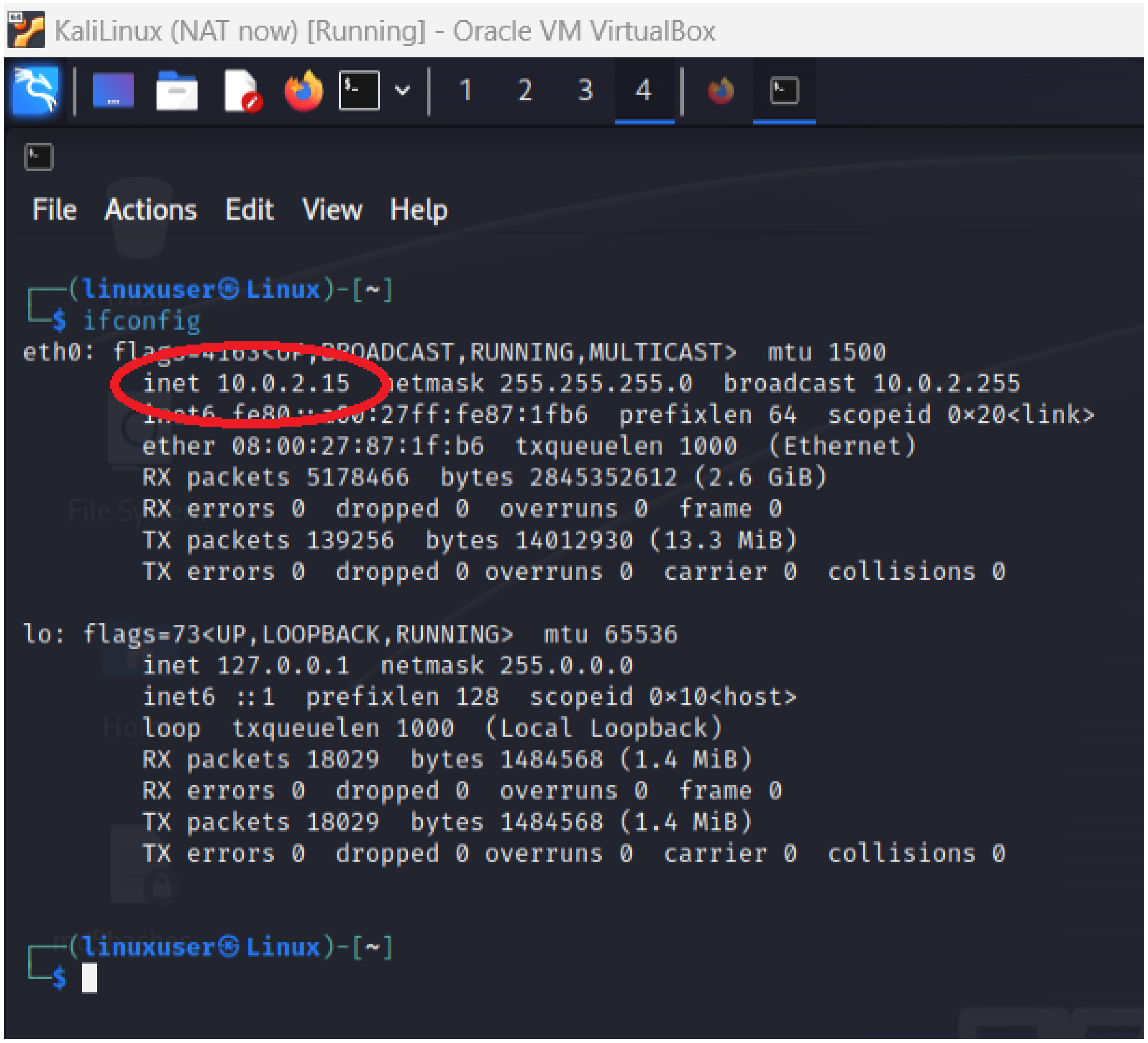
14. So first I turn on my Linux Mint Vm, and run the command: **ifconfig**, this command will display our **IP Address** which is **10.0.2.15**, since I am yet to set up a **DHCP server(Host Machine)** so that server will assign my VMs IP Addresses.

Note: The reason why I did not take a screenshot for both of these Vms side by side is because it would not fit in this document and would cause the screenshot to look really blurry therefore I separated both screenshots.



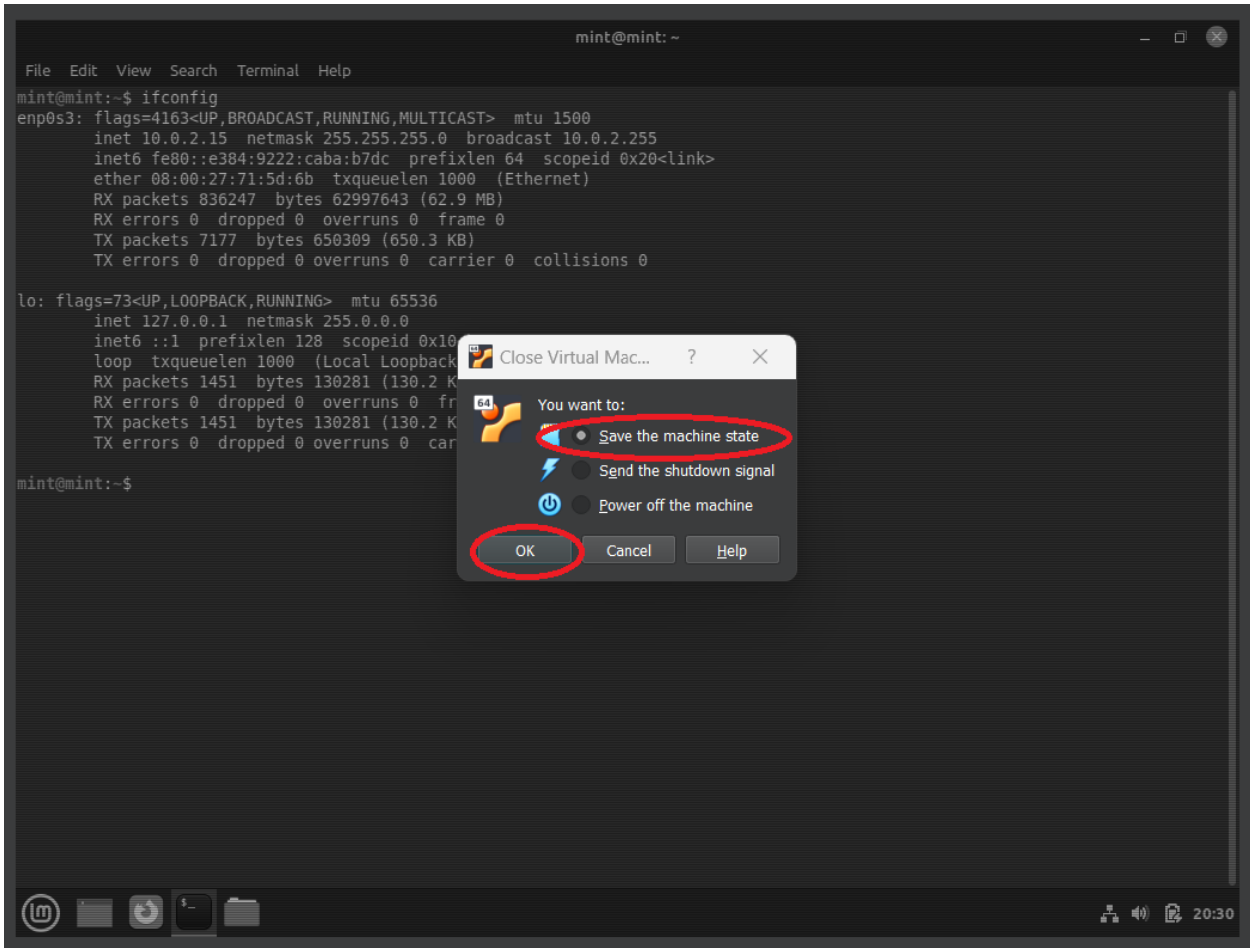
```
mint@mint:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a584:9222:caba:b7dc  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:71:5d:6b  txqueuelen 1000  (Ethernet)
        RX packets 836247  bytes 62997643 (62.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7177  bytes 650309 (650.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1451  bytes 130281 (130.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1451  bytes 130281 (130.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mint@mint:~$
```

15. Next I power on my Kali Linux VM, and execute the command: **ifconfig**, to check its IP Address and we can see its ip address is **10.0.2.15**. For both VMs we can say that they are both using **NAT(Network Address Translation)** since I am yet to create a **DHCP server** to assign these both VMs ip addresses.


KaliLinux (NAT now) [Running] - Oracle VM VirtualBox
1 2 3 4
File Actions Edit View Help

```
┌──(linuxuser㉿Linux)-[~]
└─$ 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:fe87:1fb6  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:87:1f:b6  txqueuelen 1000  (Ethernet)
        RX packets 5178466  bytes 2845352612 (2.6 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 139256  bytes 14012930 (13.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 18029  bytes 1484568 (1.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18029  bytes 1484568 (1.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(linuxuser㉿Linux)-[~]
└─$ 
```

16. After seeing that both Vms are using NAT(Network Address Translation), now I will close both VMs, so I simply **save the machine state** and click **OK** for Linux Mint VM.



```
mint@mint:~$ ifconfig
enp0s3: 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::e384:9222:caba:b7dc  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:71:5d:6b  txqueuelen 1000  (Ethernet)
        RX packets 836247  bytes 62997643 (62.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7177  bytes 650309 (650.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10
        loop  txqueuelen 1000  (Local Loopback
        RX packets 1451  bytes 130281 (130.2 K
        RX errors 0  dropped 0  overruns 0  fr
        TX packets 1451  bytes 130281 (130.2 K
        TX errors 0  dropped 0 overruns 0  car

mint@mint:~$
```

Close Virtual Mac...

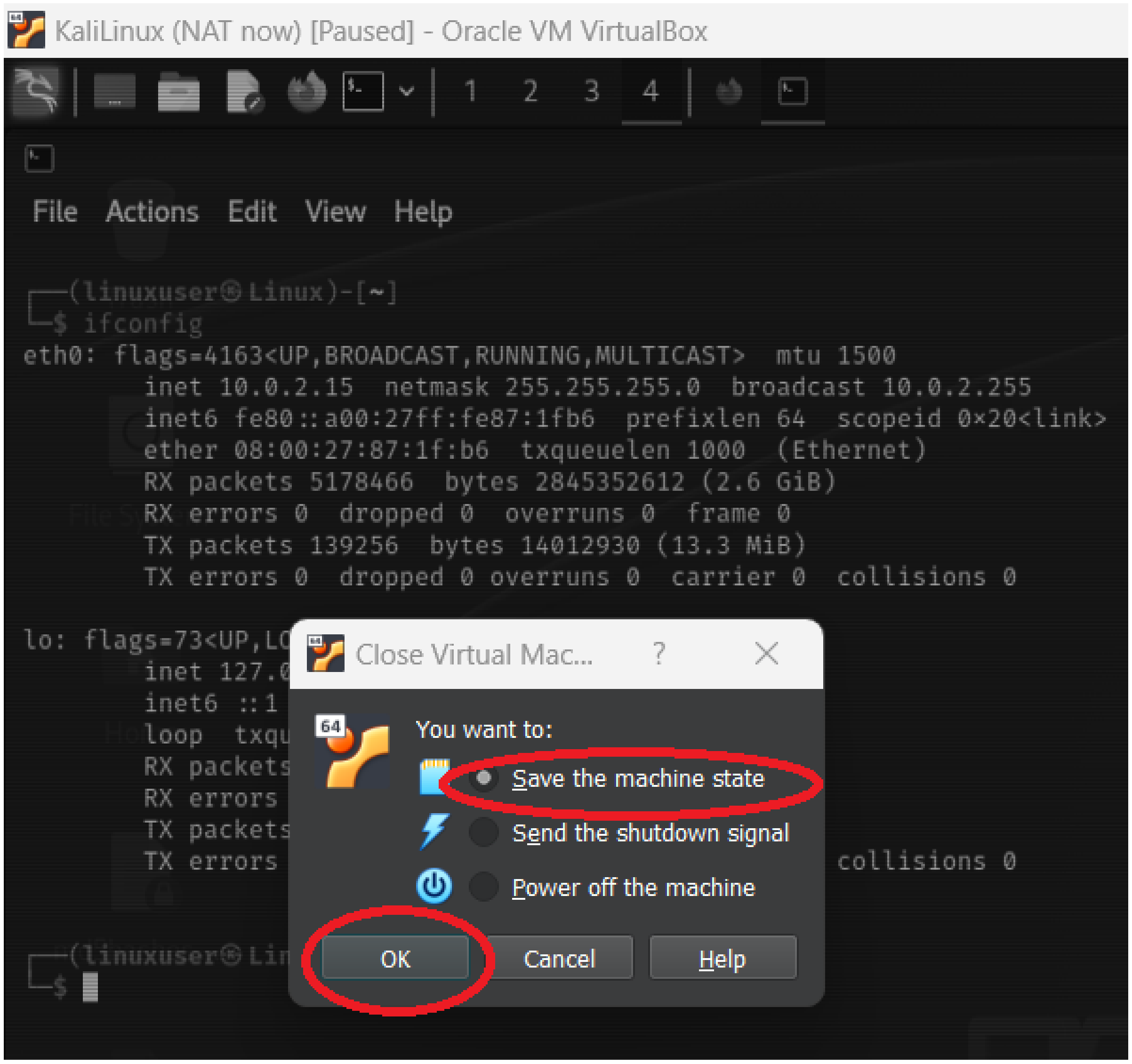
You want to:

[✓ Save the machine state]

Send the shutdown signal

Power off the machine

17. So I do the same thing for this Kali Linux VM, I save the machine state and click OK.



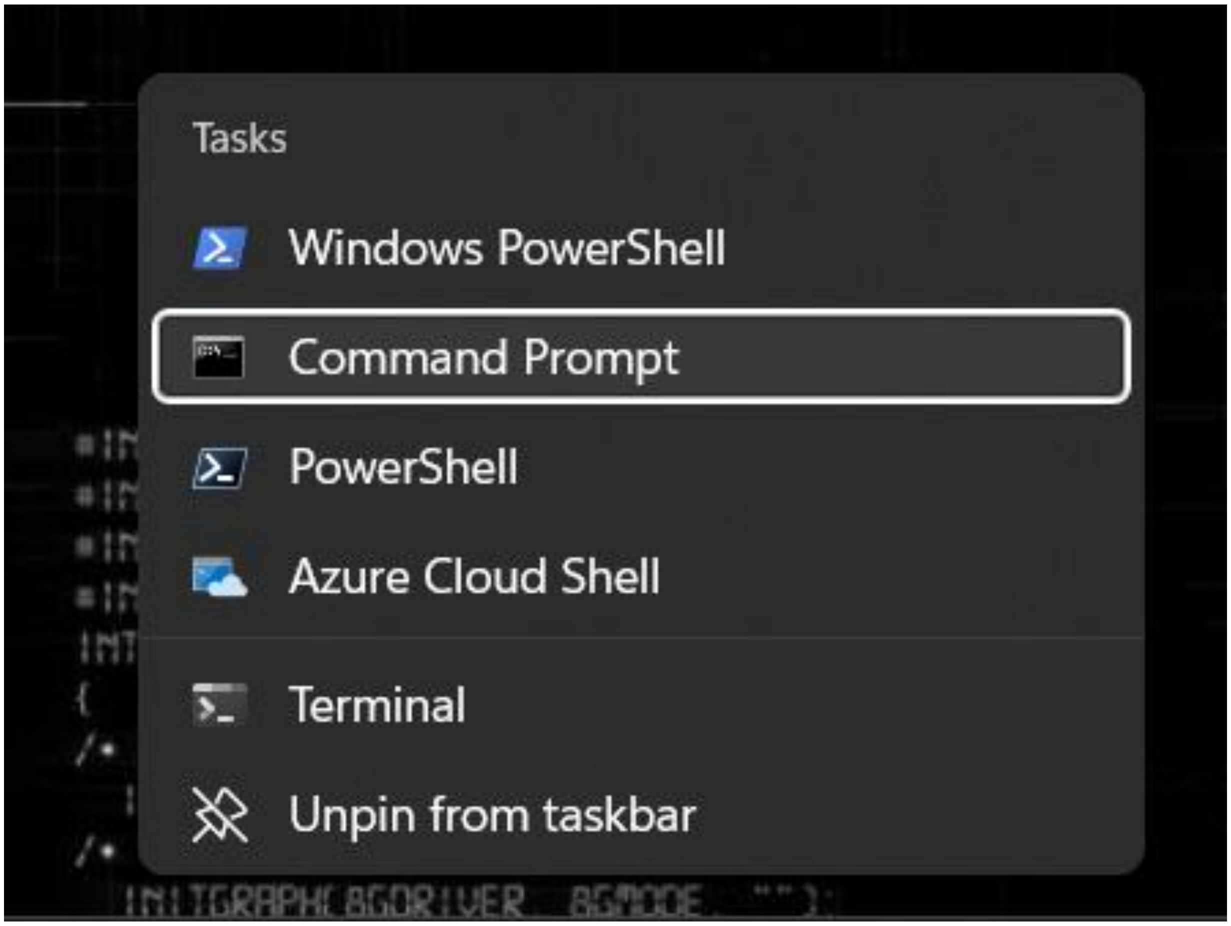
```
┌──(linuxuser㉿Linux)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe87:1fb6  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:87:1f:b6  txqueuelen 1000  (Ethernet)
        RX packets 5178466  bytes 2845352612 (2.6 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 139256  bytes 14012930 (13.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LO
        inet 127.0
        inet6 ::1
        loop  txqu
        RX packets
        RX errors
        TX packets
        TX errors                                          collisions 0

┌──(linuxuser㉿Lin
└─$ ▌
```

You want to:

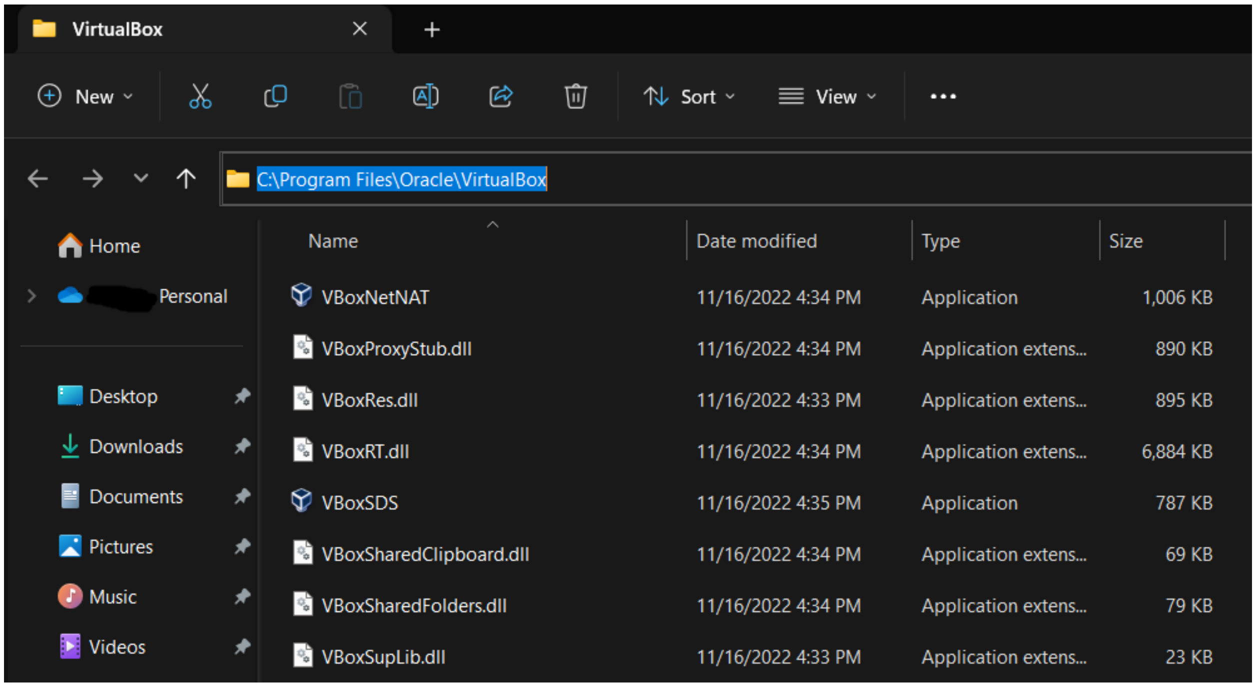
- [✓ Save the machine state]
- Send the shutdown signal
- Power off the machine

Part 2: Setting Up a DHCP Server

18. So on my host machine(Windows 11), I open up **Command Prompt** which I will be using to create my DHCP server.



19. Then I open file explorer, and navigate to **Oracle VirtualBox** and copy its **PATH.**

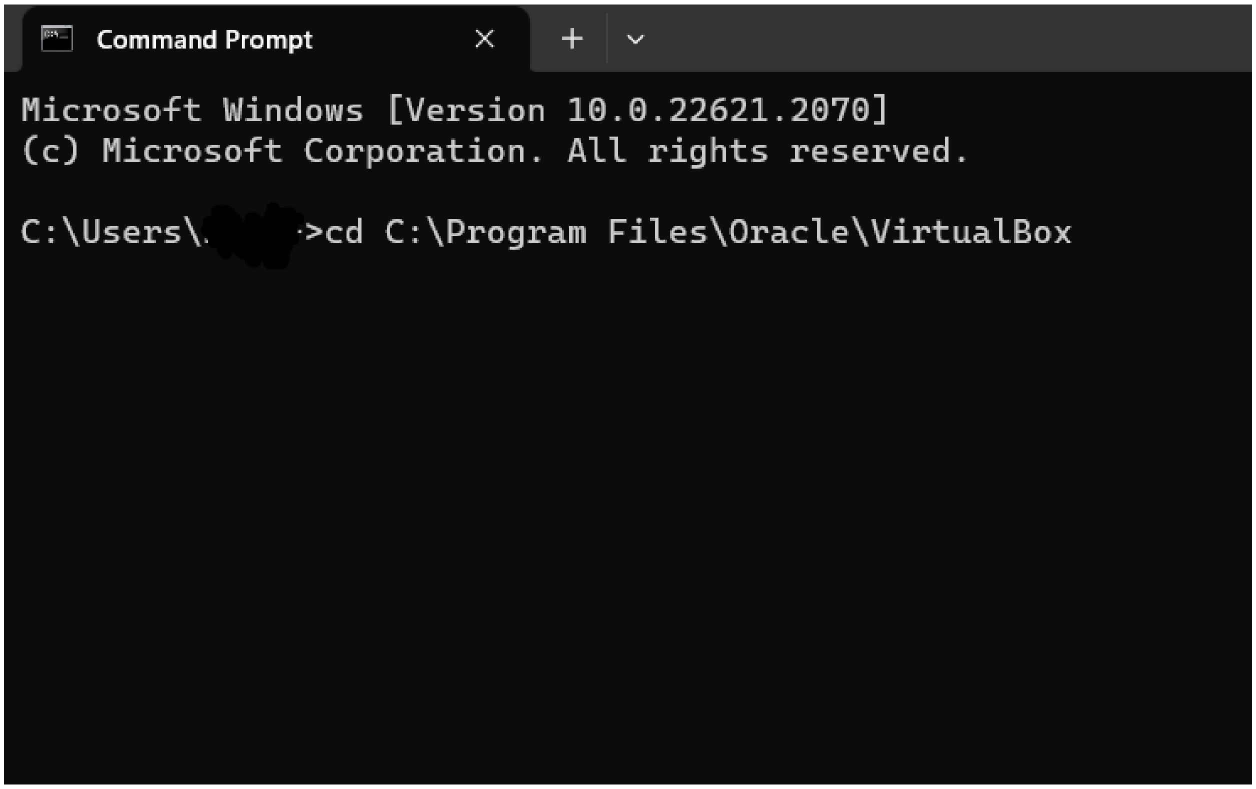


C:\Program Files\Oracle\VirtualBox

Name	Date modified	Type	Size
VBoxNetNAT	11/16/2022 4:34 PM	Application	1,006 KB
VBoxProxyStub.dll	11/16/2022 4:34 PM	Application extens...	890 KB
VBoxRes.dll	11/16/2022 4:33 PM	Application extens...	895 KB
VBoxRT.dll	11/16/2022 4:34 PM	Application extens...	6,884 KB
VBoxSDS	11/16/2022 4:35 PM	Application	787 KB
VBoxSharedClipboard.dll	11/16/2022 4:34 PM	Application extens...	69 KB
VBoxSharedFolders.dll	11/16/2022 4:34 PM	Application extens...	79 KB
VBoxSupLib.dll	11/16/2022 4:33 PM	Application extens...	23 KB

20. I resume back to the Command Prompt, and I type **cd** which stands for change directory and then paste the **PATH** of my **oracle virtualbox**.

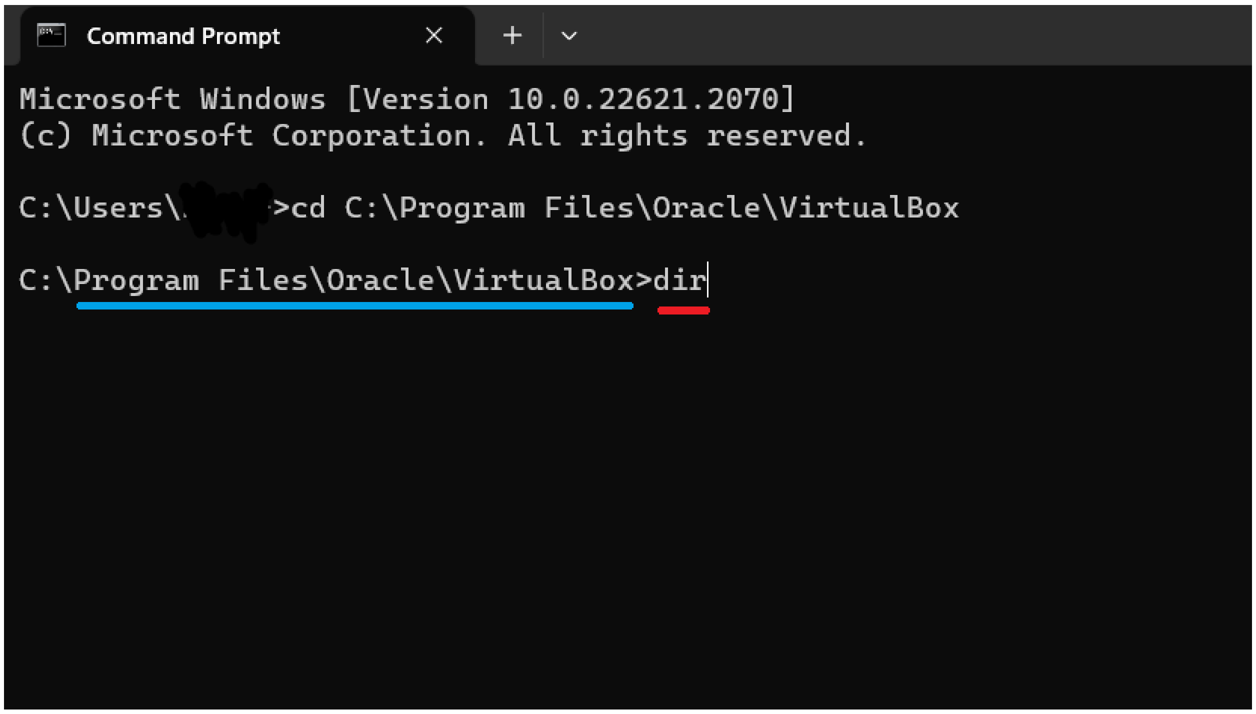
    Note: I've blurred out my host name, which explains why it directly skips to **>cd**.



```
Microsoft Windows [Version 10.0.22621.2070]
(c) Microsoft Corporation. All rights reserved.

C:\Users\[redacted]>cd C:\Program Files\Oracle\VirtualBox
```

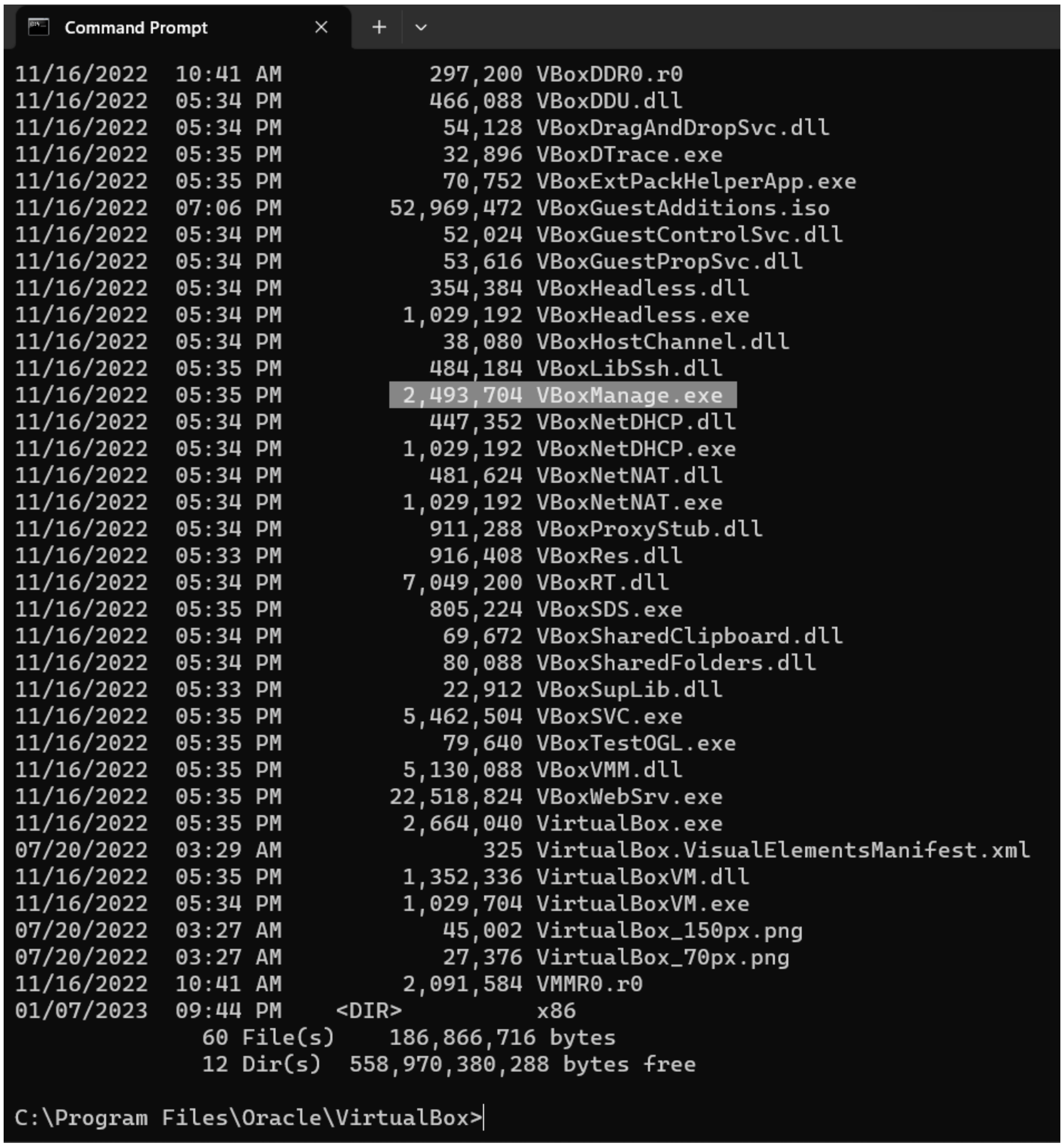
21. After executing: **cd C:\Program Files\Oracle\VirtualBox**, I now type: **dir** it will list down all the directories(folders) within our current directory.

    Note: The blue line is our **PATH** of **oracle virtualbox** and the red line is the current command I'm about to run.



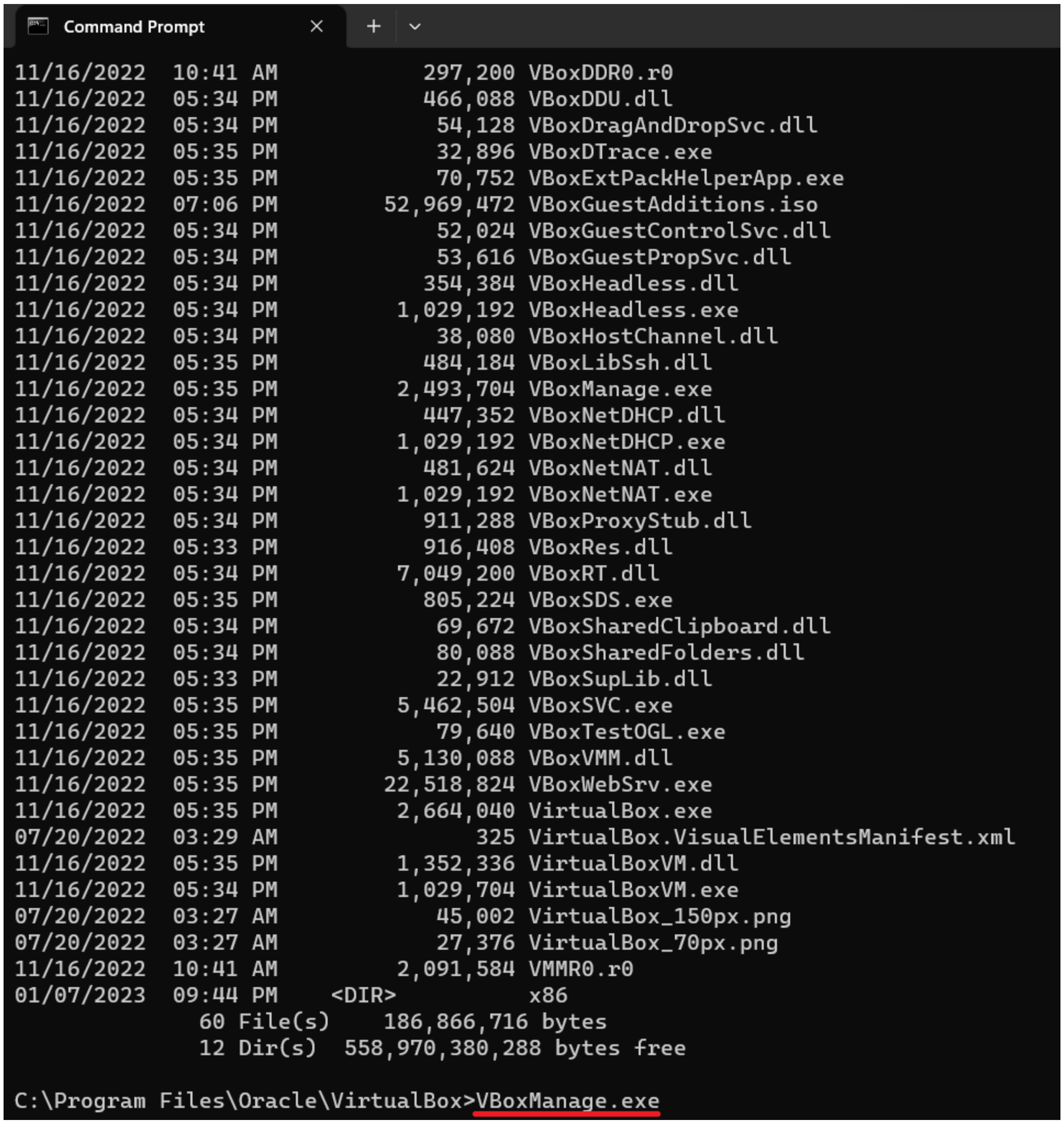
```
Microsoft Windows [Version 10.0.22621.2070]
(c) Microsoft Corporation. All rights reserved.

C:\Users\[redacted]>cd C:\Program Files\Oracle\VirtualBox

C:\Program Files\Oracle\VirtualBox>dir
```

22. After executing the command: **dir** we can now see its dynamic links and executables, and the one that I have highlighted is **VBoxManage.exe** which I will be using in the next command I run.



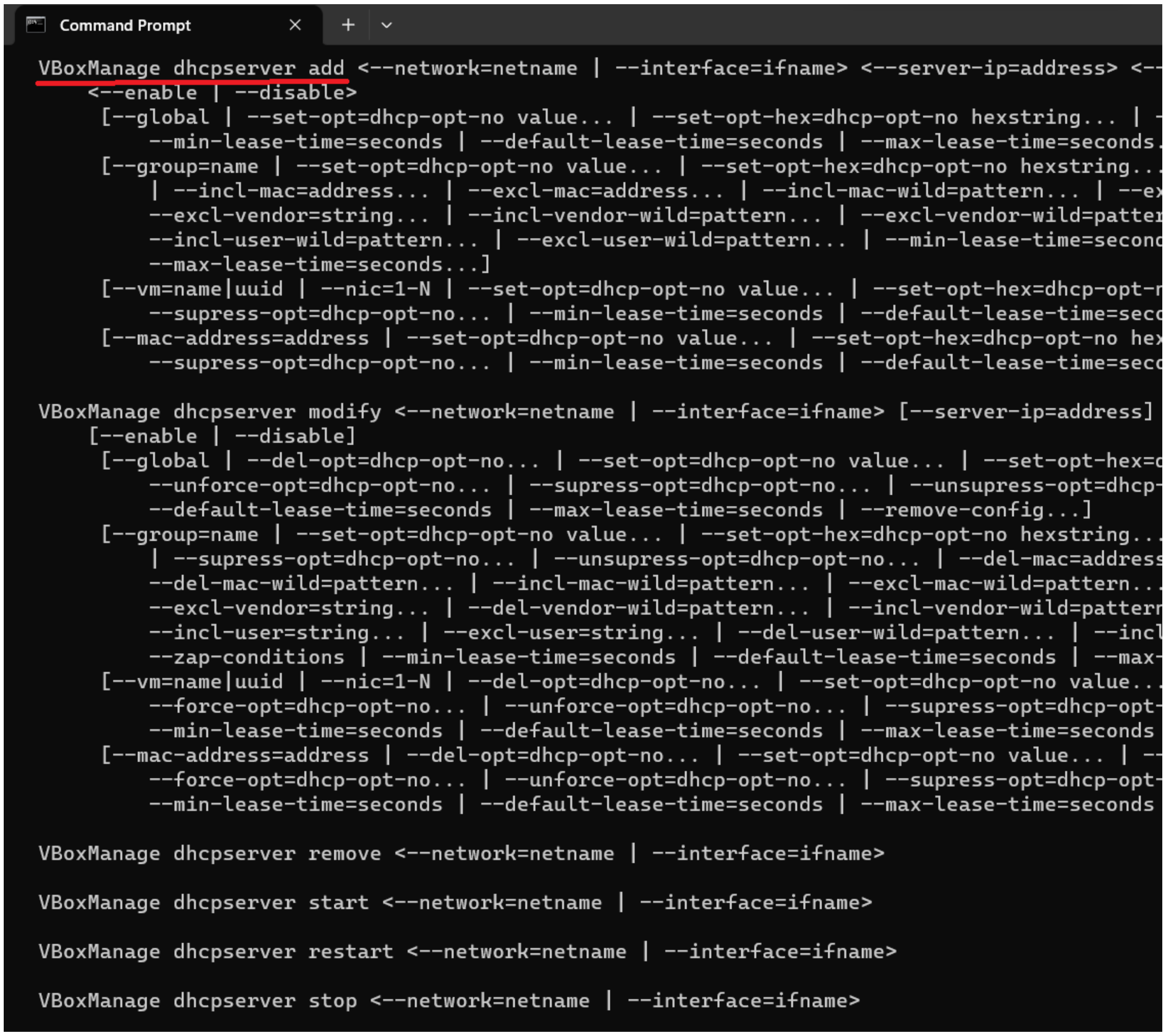
```
11/16/2022  10:41 AM           297,200 VBoxDDR0.r0
11/16/2022  05:34 PM           466,088 VBoxDDU.dll
11/16/2022  05:34 PM            54,128 VBoxDragAndDropSvc.dll
11/16/2022  05:35 PM            32,896 VBoxDTrace.exe
11/16/2022  05:35 PM            70,752 VBoxExtPackHelperApp.exe
11/16/2022  07:06 PM        52,969,472 VBoxGuestAdditions.iso
11/16/2022  05:34 PM            52,024 VBoxGuestControlSvc.dll
11/16/2022  05:34 PM            53,616 VBoxGuestPropSvc.dll
11/16/2022  05:34 PM           354,384 VBoxHeadless.dll
11/16/2022  05:34 PM         1,029,192 VBoxHeadless.exe
11/16/2022  05:34 PM            38,080 VBoxHostChannel.dll
11/16/2022  05:35 PM           484,184 VBoxLibSsh.dll
11/16/2022  05:35 PM         2,493,704 VBoxManage.exe
11/16/2022  05:34 PM           447,352 VBoxNetDHCP.dll
11/16/2022  05:34 PM         1,029,192 VBoxNetDHCP.exe
11/16/2022  05:34 PM           481,624 VBoxNetNAT.dll
11/16/2022  05:34 PM         1,029,192 VBoxNetNAT.exe
11/16/2022  05:34 PM           911,288 VBoxProxyStub.dll
11/16/2022  05:33 PM           916,408 VBoxRes.dll
11/16/2022  05:34 PM         7,049,200 VBoxRT.dll
11/16/2022  05:35 PM           805,224 VBoxSDS.exe
11/16/2022  05:34 PM            69,672 VBoxSharedClipboard.dll
11/16/2022  05:34 PM            80,088 VBoxSharedFolders.dll
11/16/2022  05:33 PM            22,912 VBoxSupLib.dll
11/16/2022  05:35 PM         5,462,504 VBoxSVC.exe
11/16/2022  05:35 PM            79,640 VBoxTestOGL.exe
11/16/2022  05:35 PM         5,130,088 VBoxVMM.dll
11/16/2022  05:35 PM        22,518,824 VBoxWebSrv.exe
11/16/2022  05:35 PM         2,664,040 VirtualBox.exe
07/20/2022  03:29 AM               325 VirtualBox.VisualElementsManifest.xml
11/16/2022  05:35 PM         1,352,336 VirtualBoxVM.dll
11/16/2022  05:34 PM         1,029,704 VirtualBoxVM.exe
07/20/2022  03:27 AM            45,002 VirtualBox_150px.png
07/20/2022  03:27 AM            27,376 VirtualBox_70px.png
11/16/2022  10:41 AM         2,091,584 VMMR0.r0
01/07/2023  09:44 PM    <DIR>          x86
              60 File(s)    186,866,716 bytes
              12 Dir(s)  558,970,380,288 bytes free

C:\Program Files\Oracle\VirtualBox>
```

23. So after checking to see if we have a **VBoxManage.exe** executable file I will now run it in the terminal.



```
11/16/2022  10:41 AM           297,200 VBoxDDR0.r0
11/16/2022  05:34 PM           466,088 VBoxDDU.dll
11/16/2022  05:34 PM            54,128 VBoxDragAndDropSvc.dll
11/16/2022  05:35 PM            32,896 VBoxDTrace.exe
11/16/2022  05:35 PM            70,752 VBoxExtPackHelperApp.exe
11/16/2022  07:06 PM        52,969,472 VBoxGuestAdditions.iso
11/16/2022  05:34 PM            52,024 VBoxGuestControlSvc.dll
11/16/2022  05:34 PM            53,616 VBoxGuestPropSvc.dll
11/16/2022  05:34 PM           354,384 VBoxHeadless.dll
11/16/2022  05:34 PM         1,029,192 VBoxHeadless.exe
11/16/2022  05:34 PM            38,080 VBoxHostChannel.dll
11/16/2022  05:35 PM           484,184 VBoxLibSsh.dll
11/16/2022  05:35 PM         2,493,704 VBoxManage.exe
11/16/2022  05:34 PM           447,352 VBoxNetDHCP.dll
11/16/2022  05:34 PM         1,029,192 VBoxNetDHCP.exe
11/16/2022  05:34 PM           481,624 VBoxNetNAT.dll
11/16/2022  05:34 PM         1,029,192 VBoxNetNAT.exe
11/16/2022  05:34 PM           911,288 VBoxProxyStub.dll
11/16/2022  05:33 PM           916,408 VBoxRes.dll
11/16/2022  05:34 PM         7,049,200 VBoxRT.dll
11/16/2022  05:35 PM           805,224 VBoxSDS.exe
11/16/2022  05:34 PM            69,672 VBoxSharedClipboard.dll
11/16/2022  05:34 PM            80,088 VBoxSharedFolders.dll
11/16/2022  05:33 PM            22,912 VBoxSupLib.dll
11/16/2022  05:35 PM         5,462,504 VBoxSVC.exe
11/16/2022  05:35 PM            79,640 VBoxTestOGL.exe
11/16/2022  05:35 PM         5,130,088 VBoxVMM.dll
11/16/2022  05:35 PM        22,518,824 VBoxWebSrv.exe
11/16/2022  05:35 PM         2,664,040 VirtualBox.exe
07/20/2022  03:29 AM               325 VirtualBox.VisualElementsManifest.xml
11/16/2022  05:35 PM         1,352,336 VirtualBoxVM.dll
11/16/2022  05:34 PM         1,029,704 VirtualBoxVM.exe
07/20/2022  03:27 AM            45,002 VirtualBox_150px.png
07/20/2022  03:27 AM            27,376 VirtualBox_70px.png
11/16/2022  10:41 AM         2,091,584 VMMR0.r0
01/07/2023  09:44 PM    <DIR>          x86
              60 File(s)    186,866,716 bytes
              12 Dir(s)  558,970,380,288 bytes free

C:\Program Files\Oracle\VirtualBox>VBoxManage.exe
```

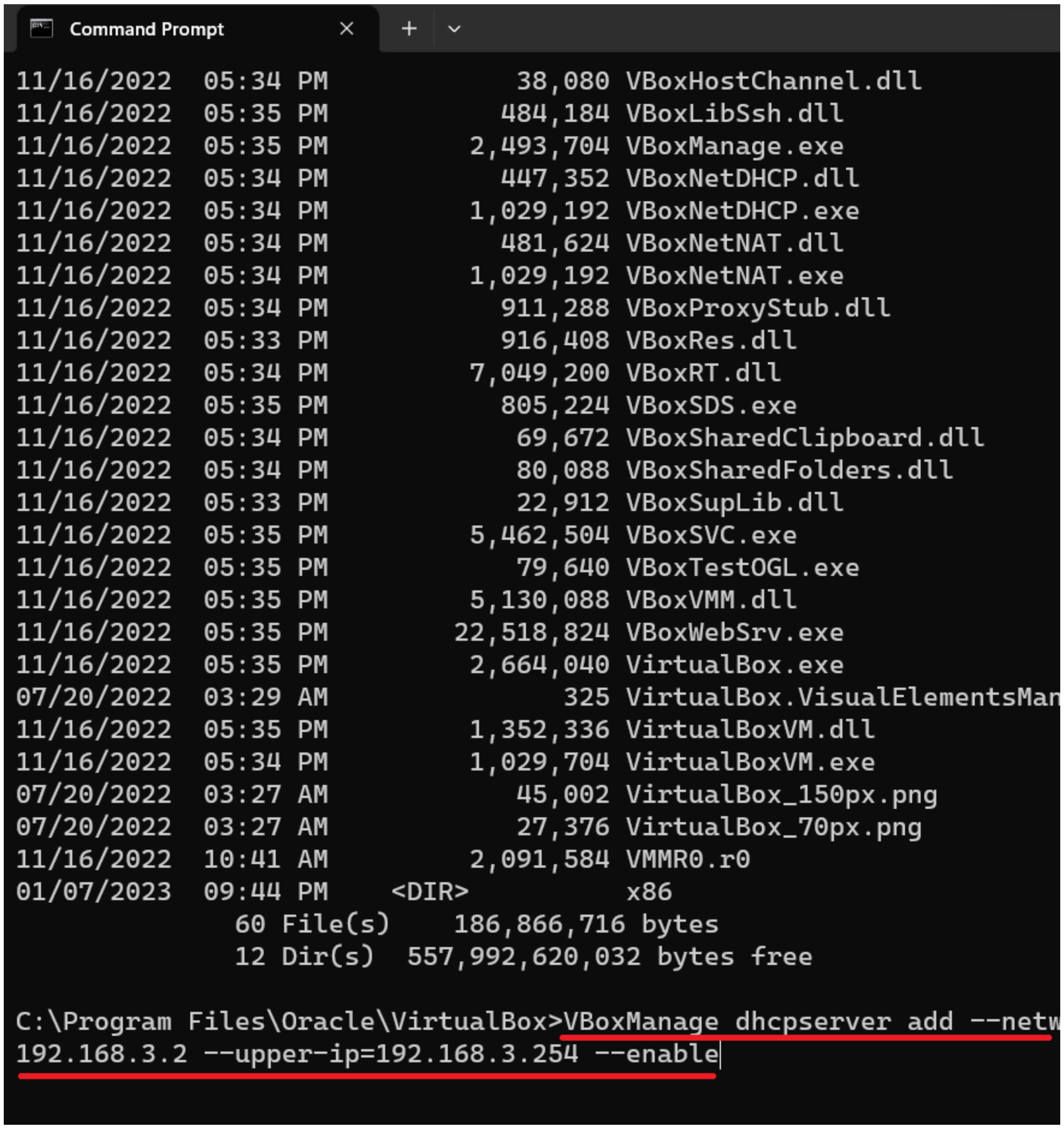
24. The previous command I've executed now displays everything we see here, so I scroll up and look for **VBoxManage dhcpserver add** which helps us know the correct syntax of adding a dhcp server.



```
VBoxManage dhcpserver add <--network=netname | --interface=ifname> <--server-ip=address> <--
    <--enable | --disable>
     [--global | --set-opt=dhcp-opt-no value... | --set-opt-hex=dhcp-opt-no hexstring... | -
         --min-lease-time=seconds | --default-lease-time=seconds | --max-lease-time=seconds.
     [--group=name | --set-opt=dhcp-opt-no value... | --set-opt-hex=dhcp-opt-no hexstring..
         | --incl-mac=address... | --excl-mac=address... | --incl-mac-wild=pattern... | --ex
         --excl-vendor=string... | --incl-vendor-wild=pattern... | --excl-vendor-wild=patter
         --incl-user-wild=pattern... | --excl-user-wild=pattern... | --min-lease-time=secon
         --max-lease-time=seconds...]
     [--vm=name|uuid | --nic=1-N | --set-opt=dhcp-opt-no value... | --set-opt-hex=dhcp-opt-n
         --supress-opt=dhcp-opt-no... | --min-lease-time=seconds | --default-lease-time=sec
     [--mac-address=address | --set-opt=dhcp-opt-no value... | --set-opt-hex=dhcp-opt-no hex
         --supress-opt=dhcp-opt-no... | --min-lease-time=seconds | --default-lease-time=sec

VBoxManage dhcpserver modify <--network=netname | --interface=ifname> [--server-ip=address]
    [--enable | --disable]
     [--global | --del-opt=dhcp-opt-no... | --set-opt=dhcp-opt-no value... | --set-opt-hex=d
         --unforce-opt=dhcp-opt-no... | --supress-opt=dhcp-opt-no... | --unsupress-opt=dhcp-
         --default-lease-time=seconds | --max-lease-time=seconds | --remove-config...]
     [--group=name | --set-opt=dhcp-opt-no value... | --set-opt-hex=dhcp-opt-no hexstring..
         | --supress-opt=dhcp-opt-no... | --unsupress-opt=dhcp-opt-no... | --del-mac=address
         --del-mac-wild=pattern... | --incl-mac-wild=pattern... | --excl-mac-wild=pattern..
         --excl-vendor=string... | --del-vendor-wild=pattern... | --incl-vendor-wild=patter
         --incl-user=string... | --excl-user=string... | --del-user-wild=pattern... | --incl
         --zap-conditions | --min-lease-time=seconds | --default-lease-time=seconds | --max-
     [--vm=name|uuid | --nic=1-N | --del-opt=dhcp-opt-no... | --set-opt=dhcp-opt-no value..
         --force-opt=dhcp-opt-no... | --unforce-opt=dhcp-opt-no... | --supress-opt=dhcp-opt-
         --min-lease-time=seconds | --default-lease-time=seconds | --max-lease-time=seconds
     [--mac-address=address | --del-opt=dhcp-opt-no... | --set-opt=dhcp-opt-no value... | --
         --force-opt=dhcp-opt-no... | --unforce-opt=dhcp-opt-no... | --supress-opt=dhcp-opt-
         --min-lease-time=seconds | --default-lease-time=seconds | --max-lease-time=seconds

VBoxManage dhcpserver remove <--network=netname | --interface=ifname>

VBoxManage dhcpserver start <--network=netname | --interface=ifname>

VBoxManage dhcpserver restart <--network=netname | --interface=ifname>

VBoxManage dhcpserver stop <--network=netname | --interface=ifname>
```

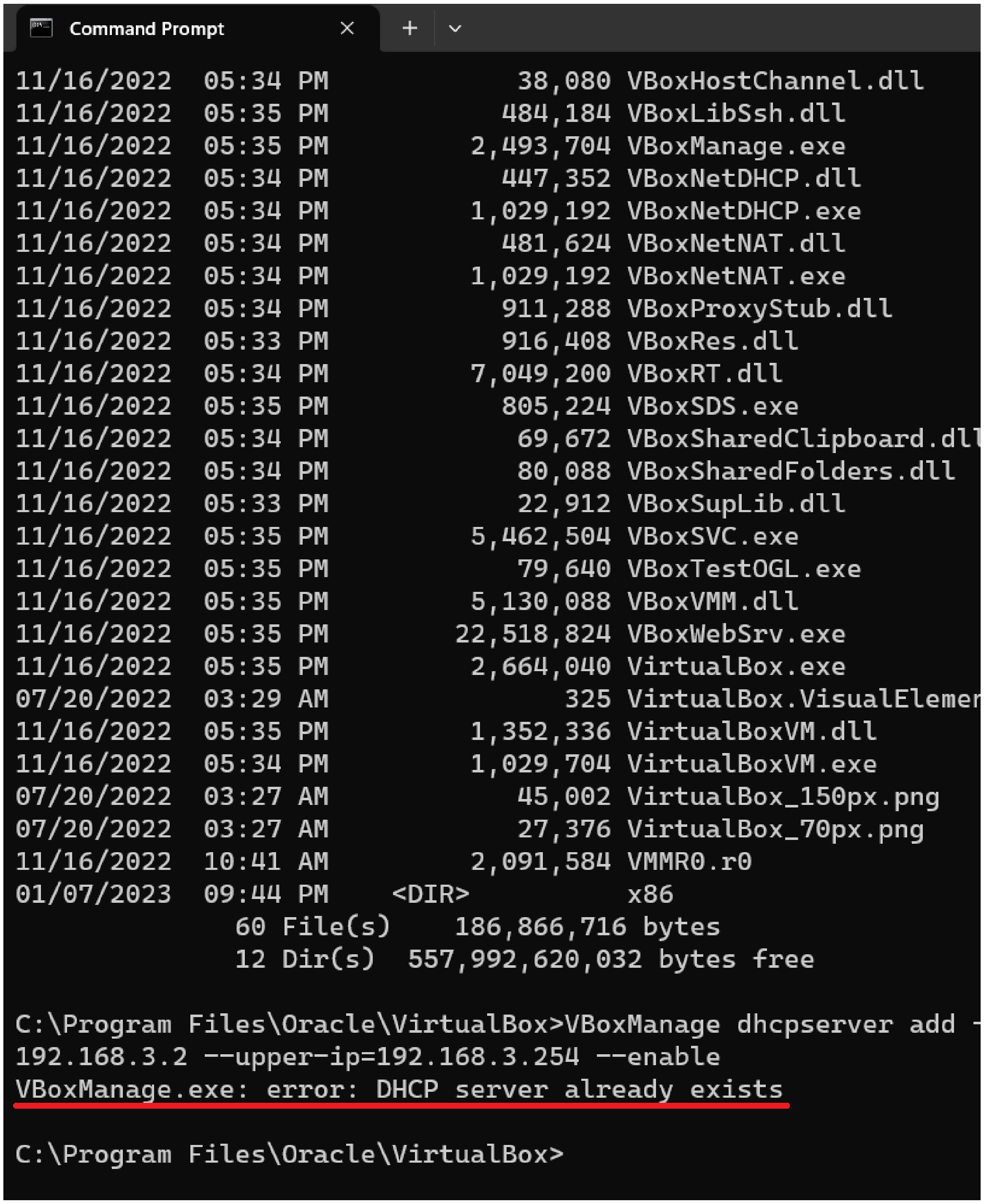
25. Using the syntax, I've seen earlier I now type my command/string: **VBoxManage dhcpserver add --network=CyberStudyLab --server-ip=192.168.3.1 --netmask=255.255.255.0 --lower-ip=192.168.3.2 --upper-ip=192.168.3.254 --enable**. Which will create a DHCP server for us to use and give our VMs ip address.

Note: For the sake of clarity I've only taken a screen of half of the command since the command is really long and taking a screenshot of that will blur the screenshot below if I did that.



```
11/16/2022  05:34 PM            38,080 VBoxHostChannel.dll
11/16/2022  05:35 PM           484,184 VBoxLibSsh.dll
11/16/2022  05:35 PM         2,493,704 VBoxManage.exe
11/16/2022  05:34 PM           447,352 VBoxNetDHCP.dll
11/16/2022  05:34 PM         1,029,192 VBoxNetDHCP.exe
11/16/2022  05:34 PM           481,624 VBoxNetNAT.dll
11/16/2022  05:34 PM         1,029,192 VBoxNetNAT.exe
11/16/2022  05:34 PM           911,288 VBoxProxyStub.dll
11/16/2022  05:33 PM           916,408 VBoxRes.dll
11/16/2022  05:34 PM         7,049,200 VBoxRT.dll
11/16/2022  05:35 PM           805,224 VBoxSDS.exe
11/16/2022  05:34 PM            69,672 VBoxSharedClipboard.dll
11/16/2022  05:34 PM            80,088 VBoxSharedFolders.dll
11/16/2022  05:33 PM            22,912 VBoxSupLib.dll
11/16/2022  05:35 PM         5,462,504 VBoxSVC.exe
11/16/2022  05:35 PM            79,640 VBoxTestOGL.exe
11/16/2022  05:35 PM         5,130,088 VBoxVMM.dll
11/16/2022  05:35 PM        22,518,824 VBoxWebSrv.exe
11/16/2022  05:35 PM         2,664,040 VirtualBox.exe
07/20/2022  03:29 AM               325 VirtualBox.VisualElementsMan
11/16/2022  05:35 PM         1,352,336 VirtualBoxVM.dll
11/16/2022  05:34 PM         1,029,704 VirtualBoxVM.exe
07/20/2022  03:27 AM            45,002 VirtualBox_150px.png
07/20/2022  03:27 AM            27,376 VirtualBox_70px.png
11/16/2022  10:41 AM         2,091,584 VMMR0.r0
01/07/2023  09:44 PM    <DIR>          x86
              60 File(s)    186,866,716 bytes
              12 Dir(s)  557,992,620,032 bytes free

C:\Program Files\Oracle\VirtualBox>VBoxManage dhcpserver add --netw
192.168.3.2 --upper-ip=192.168.3.254 --enable
```

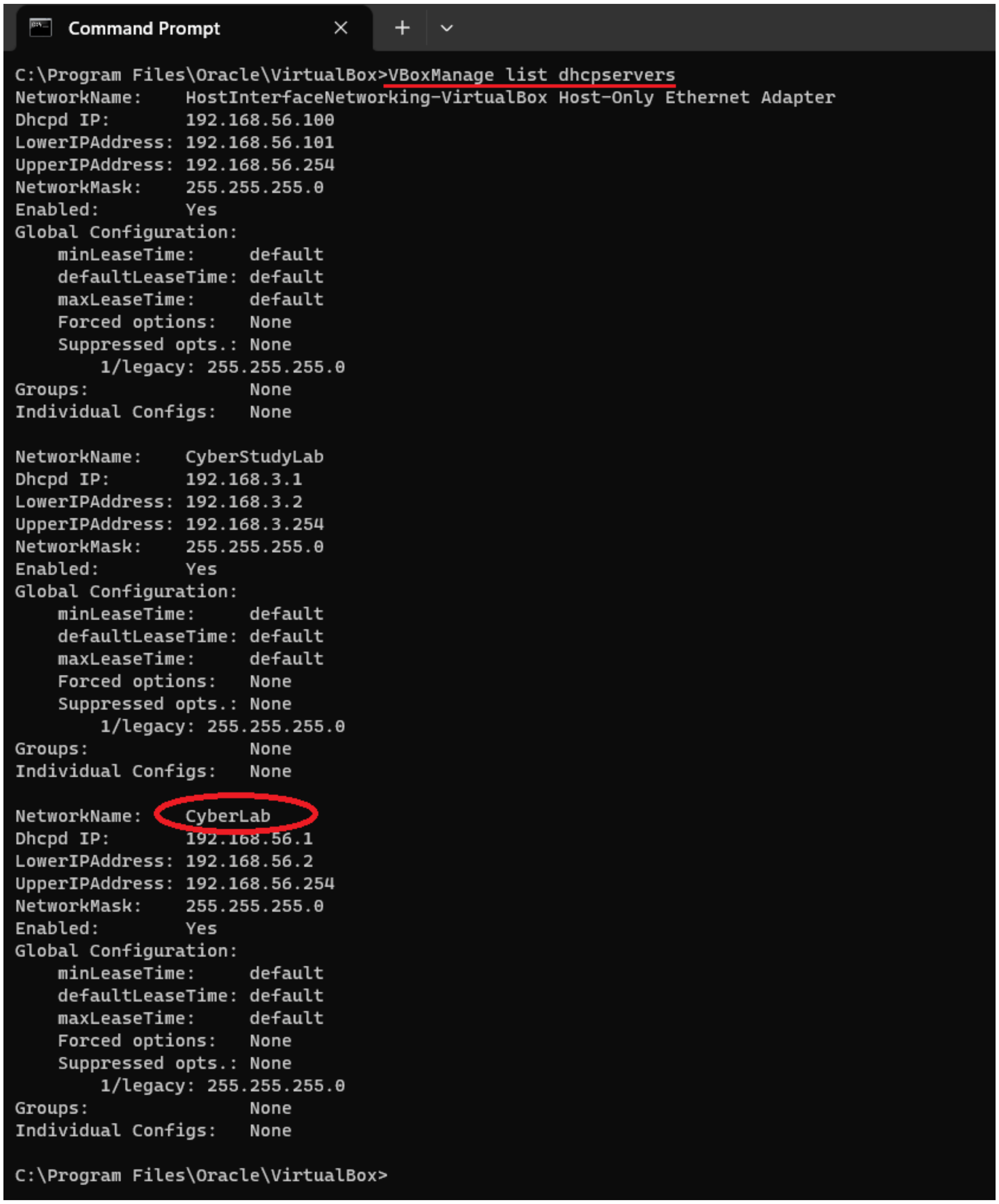
26. So after executing the really long command we can now see that I have already created a DHCP server before, which displays the error below.



Command Prompt

```
11/16/2022  05:34 PM            38,080 VBoxHostChannel.dll
11/16/2022  05:35 PM           484,184 VBoxLibSsh.dll
11/16/2022  05:35 PM         2,493,704 VBoxManage.exe
11/16/2022  05:34 PM           447,352 VBoxNetDHCP.dll
11/16/2022  05:34 PM         1,029,192 VBoxNetDHCP.exe
11/16/2022  05:34 PM           481,624 VBoxNetNAT.dll
11/16/2022  05:34 PM         1,029,192 VBoxNetNAT.exe
11/16/2022  05:34 PM           911,288 VBoxProxyStub.dll
11/16/2022  05:33 PM           916,408 VBoxRes.dll
11/16/2022  05:34 PM         7,049,200 VBoxRT.dll
11/16/2022  05:35 PM           805,224 VBoxSDS.exe
11/16/2022  05:34 PM            69,672 VBoxSharedClipboard.dll
11/16/2022  05:34 PM            80,088 VBoxSharedFolders.dll
11/16/2022  05:33 PM            22,912 VBoxSupLib.dll
11/16/2022  05:35 PM         5,462,504 VBoxSVC.exe
11/16/2022  05:35 PM            79,640 VBoxTestOGL.exe
11/16/2022  05:35 PM         5,130,088 VBoxVMM.dll
11/16/2022  05:35 PM        22,518,824 VBoxWebSrv.exe
11/16/2022  05:35 PM         2,664,040 VirtualBox.exe
07/20/2022  03:29 AM               325 VirtualBox.VisualElemen
11/16/2022  05:35 PM         1,352,336 VirtualBoxVM.dll
11/16/2022  05:34 PM         1,029,704 VirtualBoxVM.exe
07/20/2022  03:27 AM            45,002 VirtualBox_150px.png
07/20/2022  03:27 AM            27,376 VirtualBox_70px.png
11/16/2022  10:41 AM         2,091,584 VMMR0.r0
01/07/2023  09:44 PM    <DIR>          x86
              60 File(s)    186,866,716 bytes
              12 Dir(s)  557,992,620,032 bytes free

C:\Program Files\Oracle\VirtualBox>VBoxManage dhcpserver add -
192.168.3.2 --upper-ip=192.168.3.254 --enable
VBoxManage.exe: error: DHCP server already exists

C:\Program Files\Oracle\VirtualBox>
```

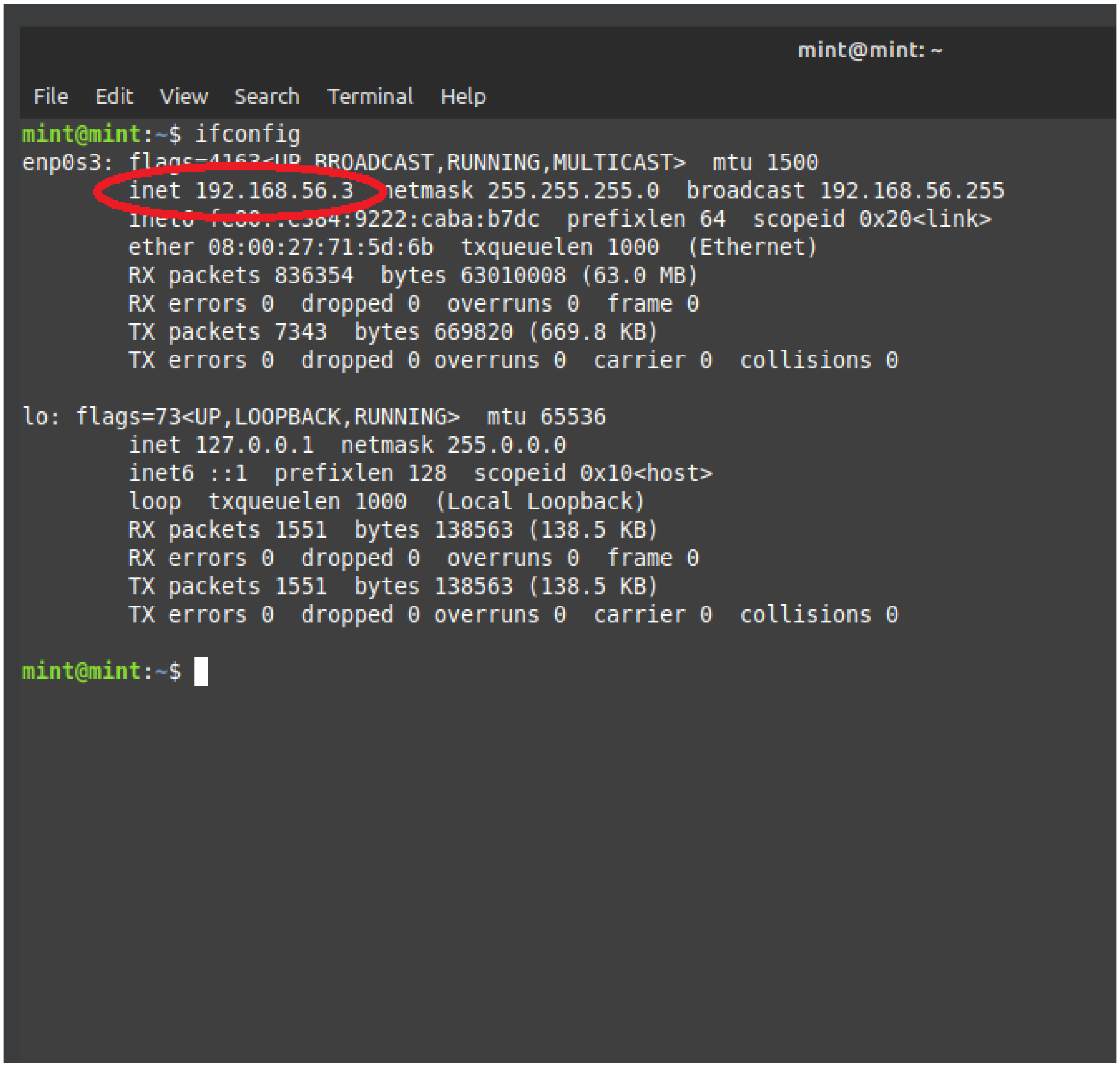
27. So to verify if I have already a DHCP server done, I execute the command: **VBoxManage list dhcpservers** and that will display the all the existing DHCP servers, in the circle I have highlighted below we can see that **CyberLab** is indeed existing in our DHCP server list



```
C:\Program Files\Oracle\VirtualBox>VBoxManage list dhcpservers
NetworkName:    HostInterfaceNetworking-VirtualBox Host-Only Ethernet Adapter
Dhcpd IP:       192.168.56.100
LowerIPAddress: 192.168.56.101
UpperIPAddress: 192.168.56.254
NetworkMask:    255.255.255.0
Enabled:        Yes
Global Configuration:
    minLeaseTime:     default
    defaultLeaseTime: default
    maxLeaseTime:     default
    Forced options:   None
    Suppressed opts.: None
        1/legacy: 255.255.255.0
Groups:               None
Individual Configs:   None

NetworkName:    CyberStudyLab
Dhcpd IP:       192.168.3.1
LowerIPAddress: 192.168.3.2
UpperIPAddress: 192.168.3.254
NetworkMask:    255.255.255.0
Enabled:        Yes
Global Configuration:
    minLeaseTime:     default
    defaultLeaseTime: default
    maxLeaseTime:     default
    Forced options:   None
    Suppressed opts.: None
        1/legacy: 255.255.255.0
Groups:               None
Individual Configs:   None

NetworkName:    CyberLab
Dhcpd IP:       192.168.56.1
LowerIPAddress: 192.168.56.2
UpperIPAddress: 192.168.56.254
NetworkMask:    255.255.255.0
Enabled:        Yes
Global Configuration:
    minLeaseTime:     default
    defaultLeaseTime: default
    maxLeaseTime:     default
    Forced options:   None
    Suppressed opts.: None
        1/legacy: 255.255.255.0
Groups:               None
Individual Configs:   None

C:\Program Files\Oracle\VirtualBox>
```

Part 3: Verifying Internal Network Connectivity

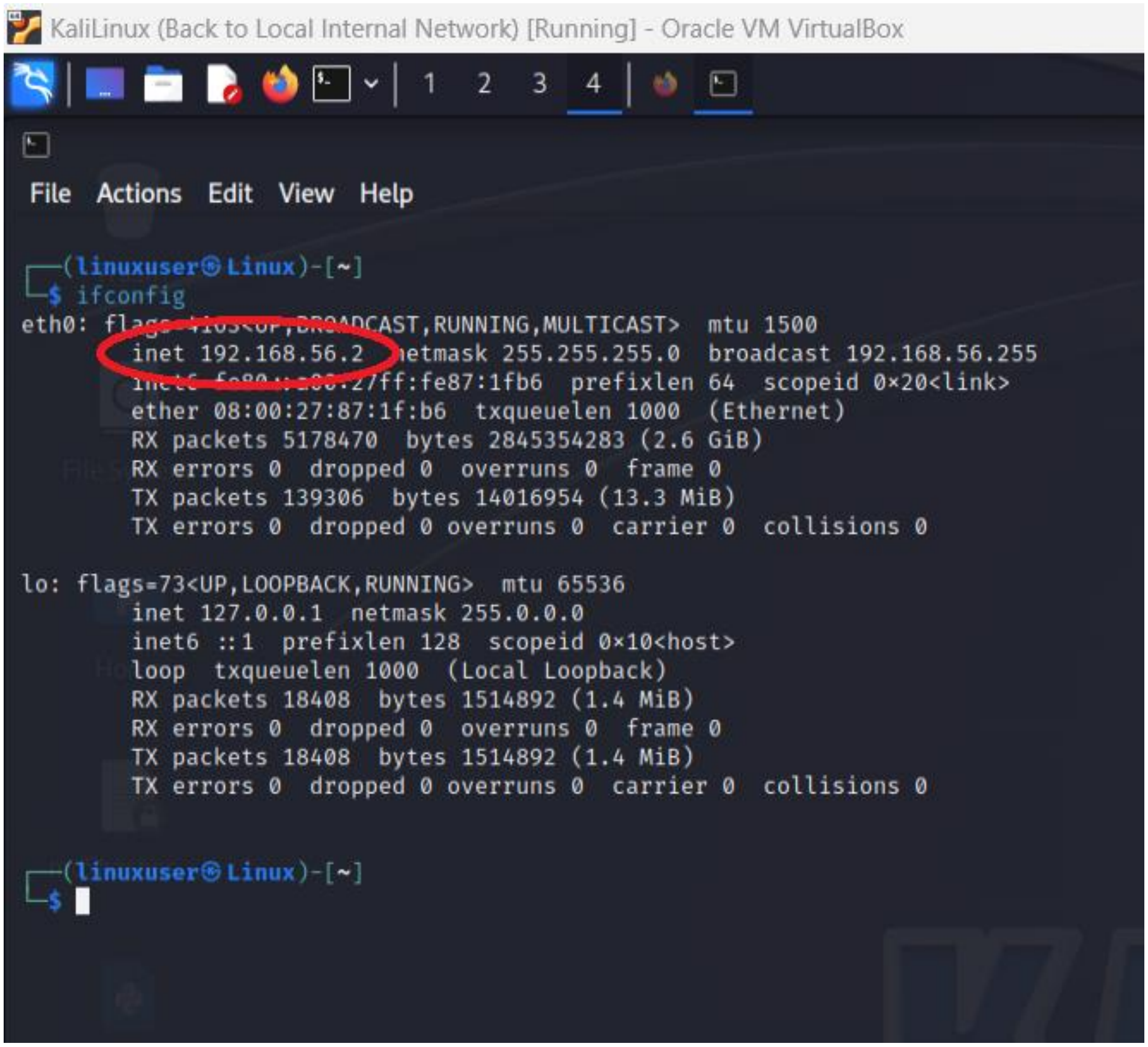
28. So I power both VMs again and I execute the command: **ifconfig** and it will display its ip address which is **192.168.56.3(Linux Mint VM).**



```
mint@mint:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.3  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::c384:9222:caba:b7dc  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:71:5d:6b  txqueuelen 1000  (Ethernet)
        RX packets 836354  bytes 63010008 (63.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7343  bytes 669820 (669.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1551  bytes 138563 (138.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1551  bytes 138563 (138.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mint@mint:~$
```

29. I do the same thing for the Kali Linux VM, I execute **ifconfig** and we can see its ip address is **192.168.56.2**. So now we can see that both VMs (Kali Linux VM, Linux Mint VM) are indeed assigned IP addresses by the dhcp server I have configured.

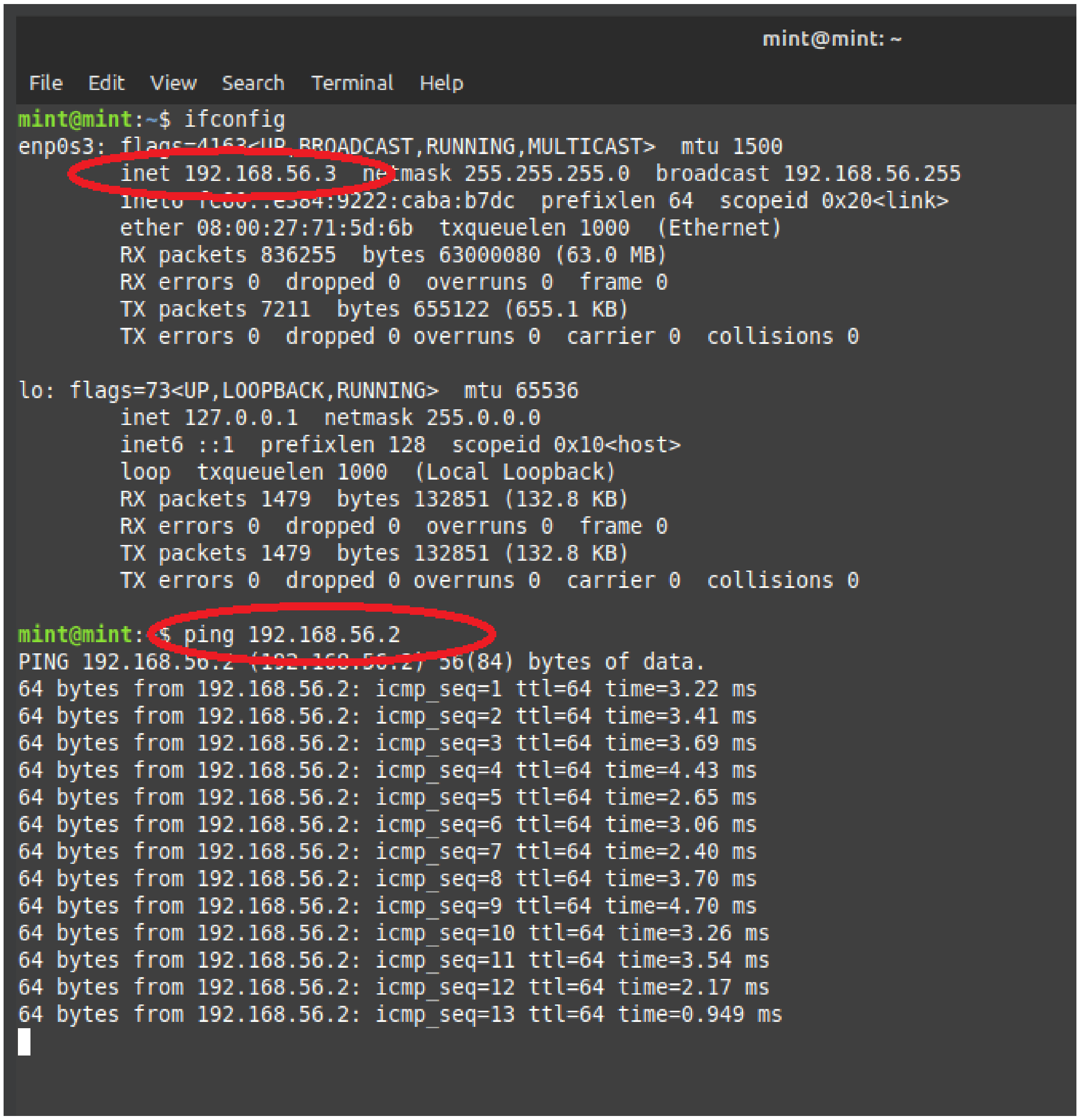
Note: The reason why I did not take a screenshot for both of these Vms side by side is because it would not fit in this document and would cause the screenshot to look really blurry therefore I separated both screenshots.


KaliLinux (Back to Local Internal Network) [Running] - Oracle VM VirtualBox
1
2
3
4
File
Actions
Edit
View
Help

```
┌──(linuxuser㉿Linux)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.2  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fe87:1fb6  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:87:1f:b6  txqueuelen 1000  (Ethernet)
        RX packets 5178470  bytes 2845354283 (2.6 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 139306  bytes 14016954 (13.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 18408  bytes 1514892 (1.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18408  bytes 1514892 (1.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(linuxuser㉿Linux)-[~]
└─$ 
```

30. So now that we have verified the DHCP server has assigned both VMs IP addresses, it's time to find out if both VMs are in the same **Internal Network**. To verify that both VMs are in the same internal network I execute the command: **ping** including the IP address of the **Kali Linux VM.** As we can see from the screenshot below, there is connectivity for both VMs.

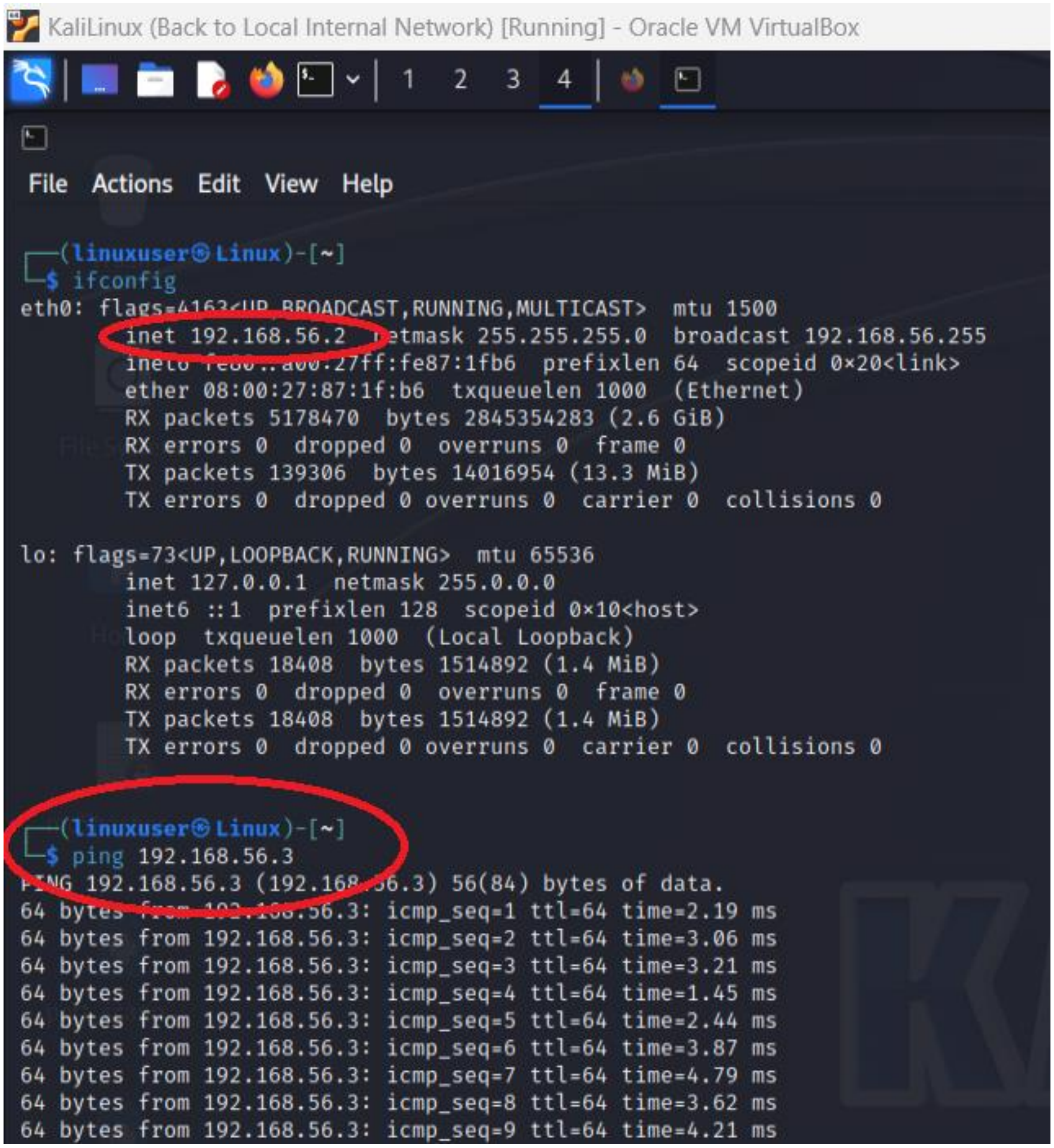
    Note: The ip address of the **Kali Linux VM** is **192.168.56.2.**
    The ip address of the **Linux Mint VM** is **192.168.56.3.**



```
mint@mint:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.3  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::e384:9222:caba:b7dc  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:71:5d:6b  txqueuelen 1000  (Ethernet)
        RX packets 836255  bytes 63000080 (63.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7211  bytes 655122 (655.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1479  bytes 132851 (132.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1479  bytes 132851 (132.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mint@mint:~$ ping 192.168.56.2
PING 192.168.56.2 (192.168.56.2) 56(84) bytes of data.
64 bytes from 192.168.56.2: icmp_seq=1 ttl=64 time=3.22 ms
64 bytes from 192.168.56.2: icmp_seq=2 ttl=64 time=3.41 ms
64 bytes from 192.168.56.2: icmp_seq=3 ttl=64 time=3.69 ms
64 bytes from 192.168.56.2: icmp_seq=4 ttl=64 time=4.43 ms
64 bytes from 192.168.56.2: icmp_seq=5 ttl=64 time=2.65 ms
64 bytes from 192.168.56.2: icmp_seq=6 ttl=64 time=3.06 ms
64 bytes from 192.168.56.2: icmp_seq=7 ttl=64 time=2.40 ms
64 bytes from 192.168.56.2: icmp_seq=8 ttl=64 time=3.70 ms
64 bytes from 192.168.56.2: icmp_seq=9 ttl=64 time=4.70 ms
64 bytes from 192.168.56.2: icmp_seq=10 ttl=64 time=3.26 ms
64 bytes from 192.168.56.2: icmp_seq=11 ttl=64 time=3.54 ms
64 bytes from 192.168.56.2: icmp_seq=12 ttl=64 time=2.17 ms
64 bytes from 192.168.56.2: icmp_seq=13 ttl=64 time=0.949 ms
```

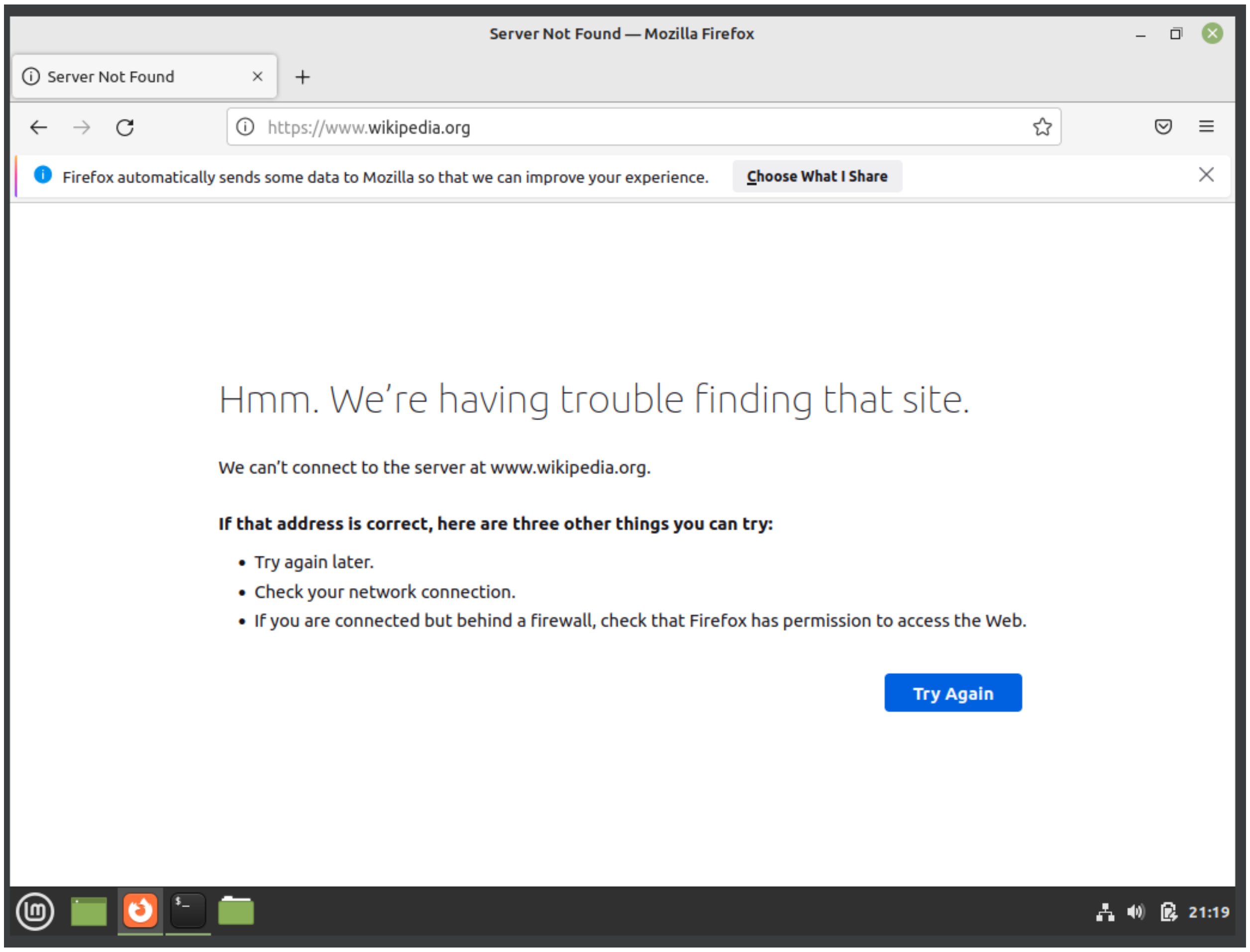
31. So now in the kali linux VM, we also want to verify that our **kali linux VM** can ping our **Linux Mint VM** by running the command **ping** and ip address of the linux mint VM which is **192.168.56.3.** So as we can see from the image below there is connectivity between both VMs therefore we can say that they are in their own internal network.

    Note: The ip address of the **Kali Linux VM** is **192.168.56.2.**
    The ip address of the **Linux Mint VM** is **192.168.56.3.**


KaliLinux (Back to Local Internal Network) [Running] - Oracle VM VirtualBox
1 2 3 4
File Actions Edit View Help

```
┌──(linuxuser㉿Linux)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.2  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fe87:1fb6  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:87:1f:b6  txqueuelen 1000  (Ethernet)
        RX packets 5178470  bytes 2845354283 (2.6 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 139306  bytes 14016954 (13.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 18408  bytes 1514892 (1.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18408  bytes 1514892 (1.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(linuxuser㉿Linux)-[~]
└─$ ping 192.168.56.3
PING 192.168.56.3 (192.168.56.3) 56(84) bytes of data.
64 bytes from 192.168.56.3: icmp_seq=1 ttl=64 time=2.19 ms
64 bytes from 192.168.56.3: icmp_seq=2 ttl=64 time=3.06 ms
64 bytes from 192.168.56.3: icmp_seq=3 ttl=64 time=3.21 ms
64 bytes from 192.168.56.3: icmp_seq=4 ttl=64 time=1.45 ms
64 bytes from 192.168.56.3: icmp_seq=5 ttl=64 time=2.44 ms
64 bytes from 192.168.56.3: icmp_seq=6 ttl=64 time=3.87 ms
64 bytes from 192.168.56.3: icmp_seq=7 ttl=64 time=4.79 ms
64 bytes from 192.168.56.3: icmp_seq=8 ttl=64 time=3.62 ms
64 bytes from 192.168.56.3: icmp_seq=9 ttl=64 time=4.21 ms
```

32. Since both VMs are in the same **internal network**, we also want to verify that they do not have any internet access, and we can verify that by opening a browser and typing a **URL** to see if it can access it, since it cannot access the **URL** we have typed we can then say that our VMs do not have internet access.


Server Not Found — Mozilla Firefox
Server Not Found
https://www.wikipedia.org
Firefox automatically sends some data to Mozilla so that we can improve your experience.
Choose What I Share
Try Again
21:19

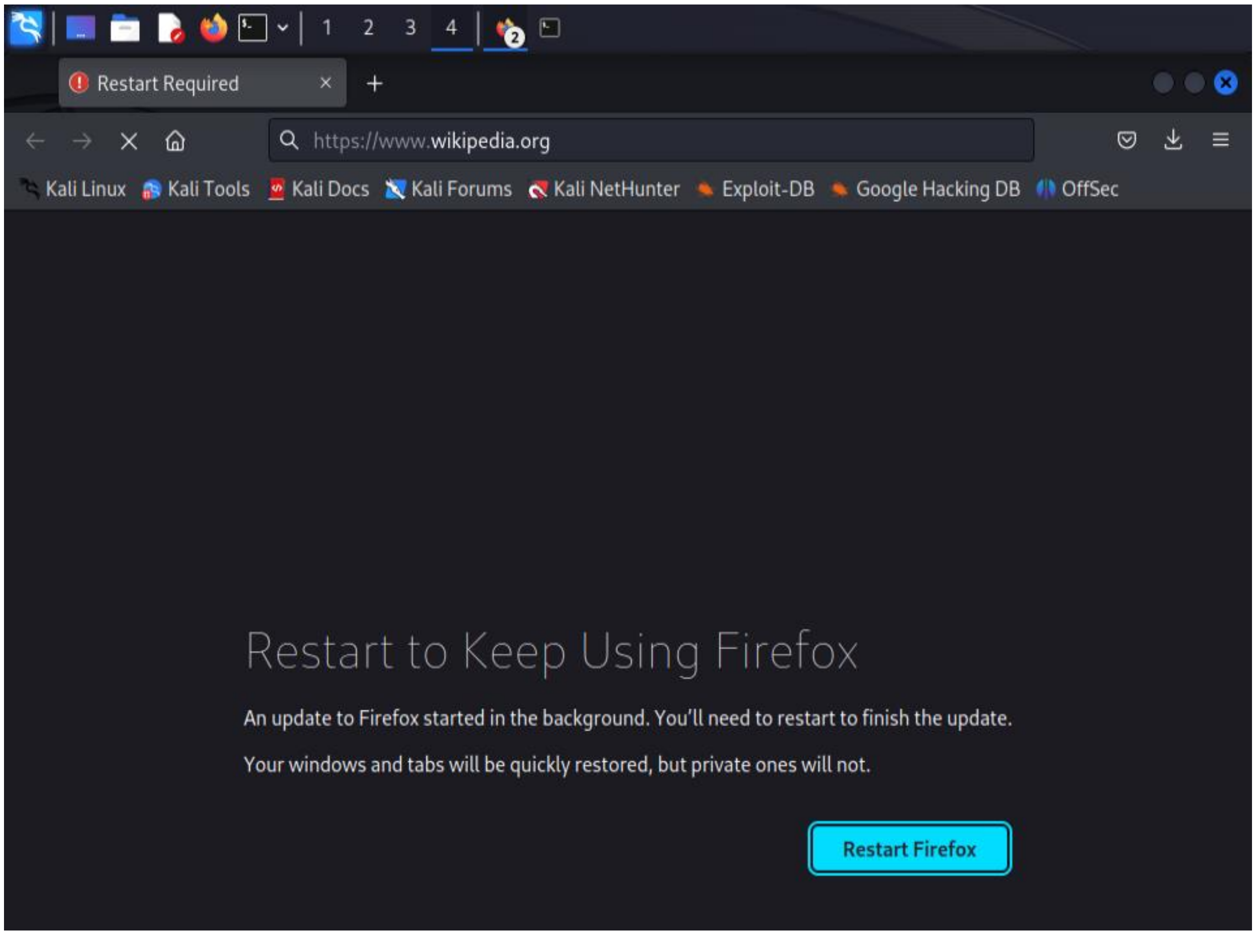
Hmm. We’re having trouble finding that site.

We can’t connect to the server at www.wikipedia.org.

If that address is correct, here are three other things you can try:

- Try again later.
- Check your network connection.
- If you are connected but behind a firewall, check that Firefox has permission to access the Web.

33. I will do the same thing for the kali linux VM, so I open the browser and type a **URL** and check to see if it can access it, since there is no access to the **URL** we can then say that there is no internet connection.



Restart to Keep Using Firefox

An update to Firefox started in the background. You'll need to restart to finish the update.

Your windows and tabs will be quickly restored, but private ones will not.

End of Documentation

References: How to build a SECURE hacking lab (VirtualBox Networking)

By: JPS