

## Appendix

# Setting Up a Penetration Testing Lab on Ubuntu Desktop

In this chapter, you will learn how to design and build a virtualized penetration testing lab environment on an Ubuntu Desktop computer and leverage virtualization technologies to reduce the cost and need to acquire multiple physical systems and devices.

In addition, you'll learn how to set up virtually isolated networks to ensure you do not accidentally target systems you do not own. Furthermore, you will set up Kali Linux as the attacker machine and Metasploitable 3 as a vulnerable system for your targets. It's important to always remember that when practicing offensive security skills such as ethical hacking and penetration testing, it should always be performed on systems and networks you own, as these security tests are usually intrusive and have the potential to cause damage to systems.

Keep in mind that you'll need to review *Chapter 2, Building a Penetration Testing Lab*, and *Chapter 3, Setting Up for Advanced Penetration Testing Techniques*, to complete the lab build.

In this chapter, we will cover the following topics:

- Setting up a hypervisor and virtual networks
- Setting up Kali Linux on Ubuntu
- Setting up Metasploitable 3 on Ubuntu

Let's dive in!

## Technical requirements

To follow along with the exercises in this chapter, please ensure that you have met the following hardware and software requirements:

- Oracle VM VirtualBox – <https://www.virtualbox.org/wiki/Downloads>

- Oracle VM VirtualBox Extension Pack – <https://www.virtualbox.org/wiki/Downloads>
- Kali Linux – <https://www.kali.org/get-kali/>
- Vagrant – <https://www.vagrantup.com/>
- Metasploitable 3 (Windows and Linux) – <https://app.vagrantup.com/rapid7>

## An overview of the lab setup and technologies used

The concept of creating your very own virtualized penetration testing lab allows you to maximize the computing resources on your existing computer, without the need to purchase online lab time from various service providers or even buy additional computers and devices. Overall, you'll be saving a lot of money as opposed to buying physical computers and networking equipment such as routers and switches.

As a cybersecurity lecturer and professional, I have noticed that many people who are starting their journeys in the field of **Information Technology (IT)** usually think that a physical lab infrastructure is needed due to their field of study. To some extent, this is true, but as technology advances, building a physical lab to practice your skills has many downsides associated with it.

The following are some of the disadvantages of a physical lab:

- Physical space is required to store the servers and networking appliances that are needed.
- The power consumption per device will result in an overall high rate of financial expenditure.
- The cost of building/purchasing each physical device is high, whether it's a network appliance or a server.

These are just some of the concerns many students and aspiring IT professionals have. In many cases, a beginner usually has a single computer such as a desktop or a laptop computer. Being able to use the virtualization technologies that have emerged as a response to these downsides has opened a multitude of doors in the field of IT. This has enabled many people and organizations to optimize and manage their hardware resources more efficiently.

In the world of virtualization, a hypervisor is a special application that allows a user to virtualize operating systems and utilizes the hardware resources on their

system so that these hardware resources can be shared with another virtualized operating system or an application. This allows you to install more than one operating system on top of your existing computer's operating system. Imagine that you are running Microsoft Windows 11 as your main operating system (commonly referred to as the *host operating system*), but you wish to run a Linux-based operating system at the same time on the same computer. You can achieve this by using a hypervisor. Hence, we are going to use virtualization to ensure we can build a cost-effective penetration testing lab environment.

When designing a penetration testing lab environment, we'll need the following components:

- **Hypervisor** – The hypervisor is an application that enables us to virtualize operating systems and run them on any hardware. We can use a hypervisor to create multiple virtual machines that can run simultaneously on our computer. There are many hypervisor applications; we'll be using **Oracle VM VirtualBox** as our preferred application because it's free and easy to use.
- **Attacker machine** – The attacker machine will be used to create and launch various types of cyber-attacks and threats to identify and exploit security vulnerabilities on targeted systems. For the attacker machine, we'll be using Kali Linux.
- **Vulnerable machines** – Without any vulnerable systems, our lab environment will not be complete. We'll set up vulnerable systems such as Metasploitable 2, which is a Linux-based operating system with hosted web applications, and Metasploitable 3 with its Windows- and Linux-based server versions. In addition, there will be a Windows Server with two Windows client machines for learning security vulnerabilities in Microsoft authentication systems.
- **Vulnerable web application** – This will help you better understand how threat actors are able to discover and exploit security weaknesses within web applications. We'll set up the **Open Web Application Security Project (OWASP) Juice Shop** web application on Kali Linux using a Docker container.
- **Internet access** – Internet connectivity will be set up on the Kali Linux virtual machine. This will be for the convenience of easily downloading additional applications, tools, and software packages.

The following diagram shows the network topology for our virtualized penetration testing lab environment:

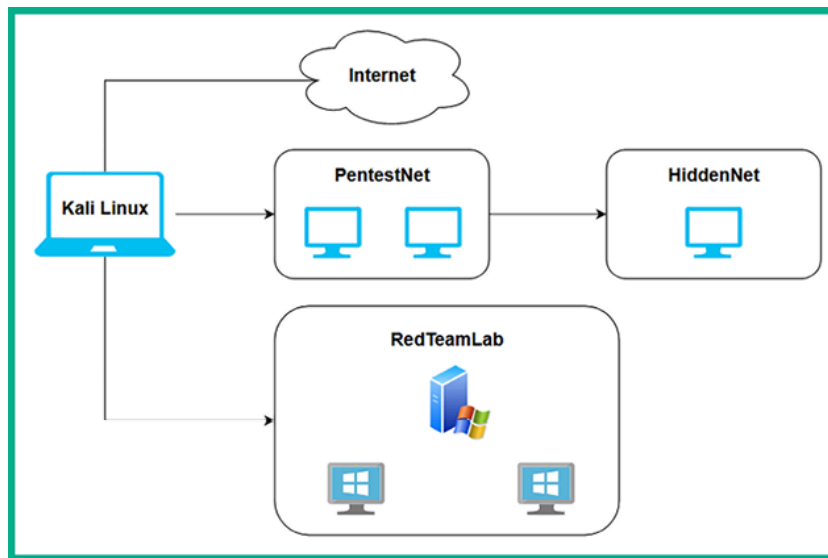


Figure 19.1: High-level network topology

As shown in the preceding diagram, there are four network zones. These are:

- The Internet for accessing online resources and is directly connected to the Kali Linux virtual machine.
- The **PentestNet** environment, which contains two vulnerable machines that are on the `172.30.1.0/24` network, and is also directly connected to Kali Linux.
- The **RedTeamLab** environment, which contains an **Active Directory (AD)** infrastructure with a Windows Server and two clients that are on the `192.168.42.0/24` network, and is directly connected to Kali Linux.
- The **HiddenNet** environment, which contains a single vulnerable host, that is, the Metasploitable 3 – Linux-based machine on the `10.11.12.0/24` network and is reachable via the **PentestNet** network only. Therefore, we'll need to compromise a host on the **PentestNet** environment and determine whether there's a way to pivot our attacks.

The following diagram provides more technical details to gain a better understanding of where specific IP networks are assigned in our lab environment:

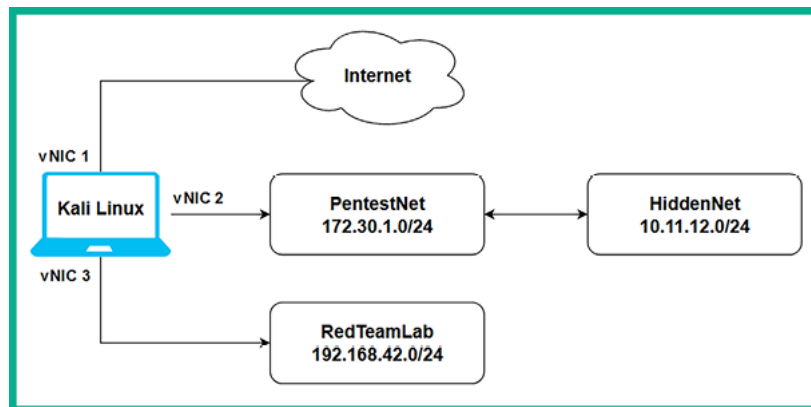


Figure 19.2: Low-level network topology

As shown in the preceding diagram, the Kali Linux virtual machine will be assigned three network adapters, which are commonly referred to as **virtual Network Interface Cards (vNICs)** on hypervisors. These vNICs enable us to access the following:

- The internet using a bridged connection
- The PentestNet environment on 172.30.1.0/24
- The RedTeamLab environment on 192.168.42.0/24

This lab design is perfect for learning how to perform lateral movement between systems, pivoting from one network to another, and compromising an AD environment.

Now that you have an idea of the virtual lab environment, as well as the systems and technologies that we are going to be working with throughout this book, let's get started with setting up the hypervisor and virtual networks next.

## Setting up a hypervisor and virtual networks

There are many hypervisors from various vendors in the information technology industry. However, Oracle VM VirtualBox is a free and simple-to-use hypervisor that has all the same essential features as commercial (paid) products. In this section, you will learn how to set up Oracle VM VirtualBox and create virtual networks on your computer.

Before getting started, the following are important factors and requirements:

- Ensure the computer's processor supports virtualization features such as **VT-x/AMD-V**.
- Ensure the virtualization feature is enabled on your processor via the BIOS/UEFI.



If you're unsure of how to access the BIOS/UEFI on your computer, please check the manual of the device or the vendor's website for specific instructions.

To get started with this exercise, please use the following instructions:

1. Open Terminal within Ubuntu Desktop and use the following commands to install Oracle VirtualBox and its extension pack:

```
glen@ubuntu:~$ sudo apt update
glen@ubuntu:~$ sudo apt install virtualbox virtualbox-ext-pack
```

2. Next, use the following commands to create the virtually isolated network using `VBoxManage` from VirtualBox:

```
glen@ubuntu:~$ cd /usr/bin/
glen@ubuntu:/usr/bin$ VBoxManage dhcpserver add --netname PentestNet --ip 172.30.1.1 --netmask 255.255.255.0 --lowerip 1
glen@ubuntu:/usr/bin$ VBoxManage dhcpserver add --netname HiddenNet --ip 10.11.12.1 --netmask 255.255.255.0 --lowerip 10
glen@ubuntu:/usr/bin$ VBoxManage dhcpserver add --netname RedTeamLab --ip 192.168.42.1 --netmask 255.255.255.0 --lowerip
```

The following screenshot shows the execution of the preceding commands:

```
glen@ubuntu:~$ cd /usr/bin/
glen@ubuntu:/usr/bin$ VBoxManage dhcpserver add --netname PentestNet --ip 172.30.1.1 --netmask 255
.255.255.0 --lowerip 172.30.1.20 --upperip 172.30.1.50 --enable
glen@ubuntu:/usr/bin$ VBoxManage dhcpserver add --netname HiddenNet --ip 10.11.12.1 --netmask 255
.255.255.0 --lowerip 10.11.12.20 --upperip 10.11.12.50 --enable
glen@ubuntu:/usr/bin$ VBoxManage dhcpserver add --netname RedTeamLab --ip 192.168.42.1 --netmask 2
55.255.255.0 --lowerip 192.168.42.20 --upperip 192.168.42.50 --set-opt=6 192.168.42.40 --enable
```

Figure 19.3: Creating virtual networks

## Setting up Kali Linux on Ubuntu

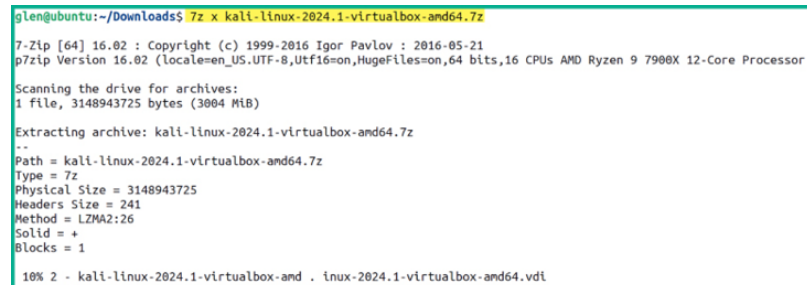
1. Open the web browser within Ubuntu, go to <https://www.kali.org/get-kali/>, and download the VirtualBox version of Kali Linux. Ensure the downloaded file is saved within your `Downloads` directory.
2. After the download is completed, use the following command to install 7-Zip, an application to unzip compressed files (Kali Linux):

```
glen@ubuntu:~$ sudo apt install p7zip-full
```

3. Next, use the following commands to change the work directory to the `Downloads` folder and unzip the file:

```
glen@ubuntu:~$ cd Downloads/  
glen@ubuntu:~/Downloads$ 7z x kali-linux-2024.1-virtualbox-amd64.7z
```

As shown in the following screenshot, 7-Zip is uncompressing the file and extracting its contents:



```
glen@ubuntu:~/Downloads$ 7z x kali-linux-2024.1-virtualbox-amd64.7z  
7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21  
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,16 CPUs AMD Ryzen 9 7900X 12-Core Processor)  
Scanning the drive for archives:  
1 file, 3148943725 bytes (3004 MiB)  
Extracting archive: kali-linux-2024.1-virtualbox-amd64.7z  
--  
Path = kali-linux-2024.1-virtualbox-amd64.7z  
Type = 7z  
Physical Size = 3148943725  
Headers Size = 241  
Method = LZMA2:26  
Solid = +  
Blocks = 1  
10% 2 - kali-linux-2024.1-virtualbox-and . inux-2024.1-virtualbox-amd64.vdi
```

Figure 19.4: Extracting file contents

4. Next, on Ubuntu Desktop, open the applications menu and click on **VirtualBox**.
5. When VirtualBox opens, click on **Add**, as shown below:

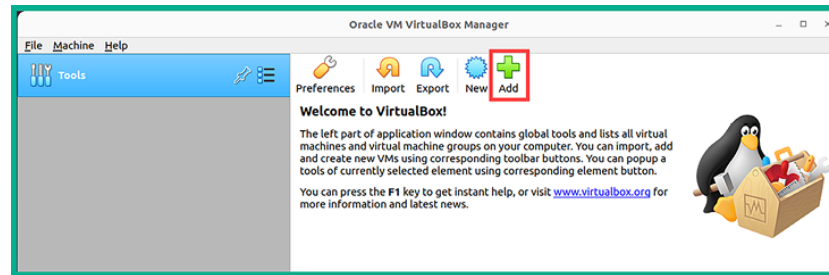


Figure 19.5: VirtualBox

6. Next, the **Select a virtual machine file** window will appear. Navigate to the **Downloads** folder, then into the extracted **kali-linux-2024.1-virtualbox-amd64** folder, and select the **kali-linux-2024.1-virtualbox-amd64.vbox** file and click on **Open**, as shown below:

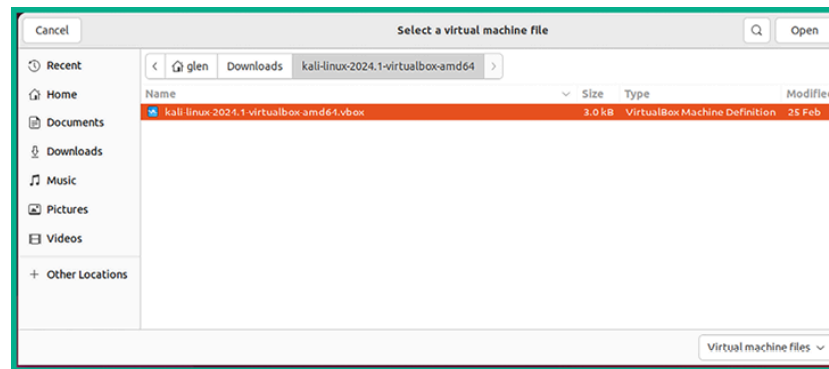


Figure 19.6: Importing Kali Linux

7. On VirtualBox, select the newly imported Kali Linux virtual machine and click on **Settings**, as shown below:



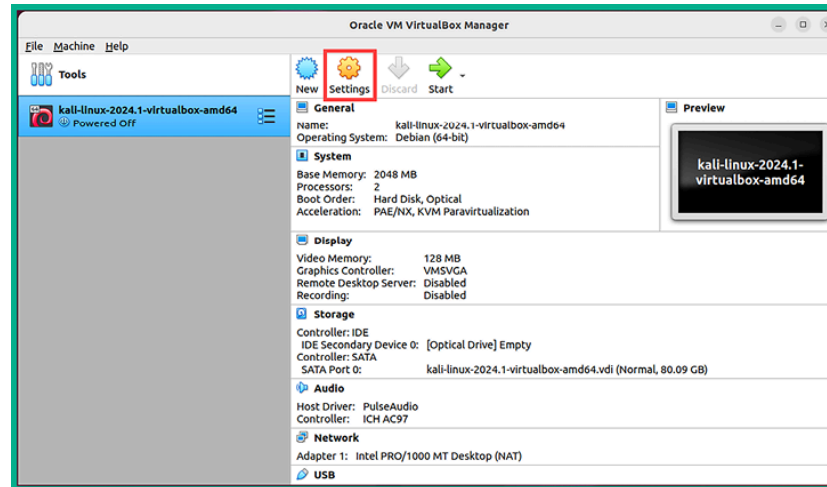


Figure 19.7: Kali Linux virtual machine.

8. Within the **Settings** menu of Kali Linux, select **Network | Adapter 1** and use the following configurations:

1. Enable the network adapter.
2. **Attached to: Bridged Adapter.**
3. **Name:** Use the drop-down menu to select the physical network adapter that's connected to your physical network with internet access.

The following screenshot shows the preceding configurations applied to **Adapter 1** (vNIC 1):

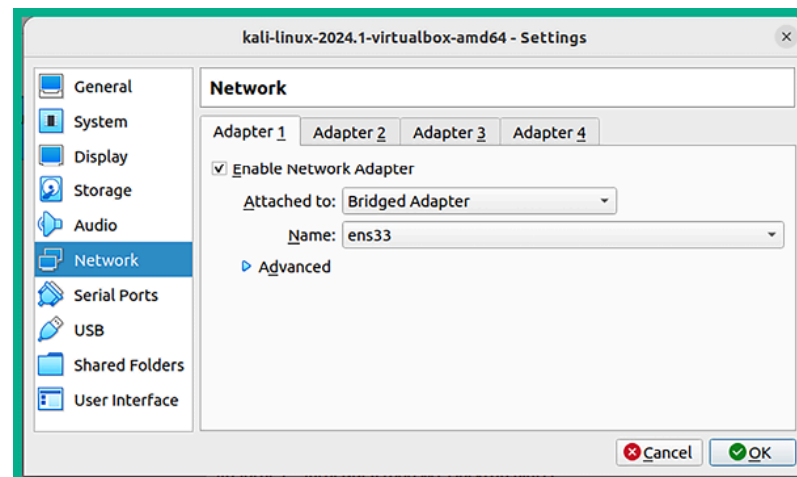


Figure 19.8: Network Adapter 1

9. Next, let's assign **Adapter 2** (vNIC 2) to the `PentestNet` network. Select the **Adapter 2** tab and use the following configurations:
1. Enable the network adapter.
  2. **Attached to: Internal Network.**
  3. **Name:** Manually enter `PentestNet` within the field.
  4. **Promiscuous Mode: Allow All.**



Enabling promiscuous mode on a network interface enables the Kali Linux machine to capture and process all the packets that the interface receives. This is good for performing packet capturing and analysis.

The following screenshot shows the preceding configurations applied to **Adapter 2** (vNIC 2):

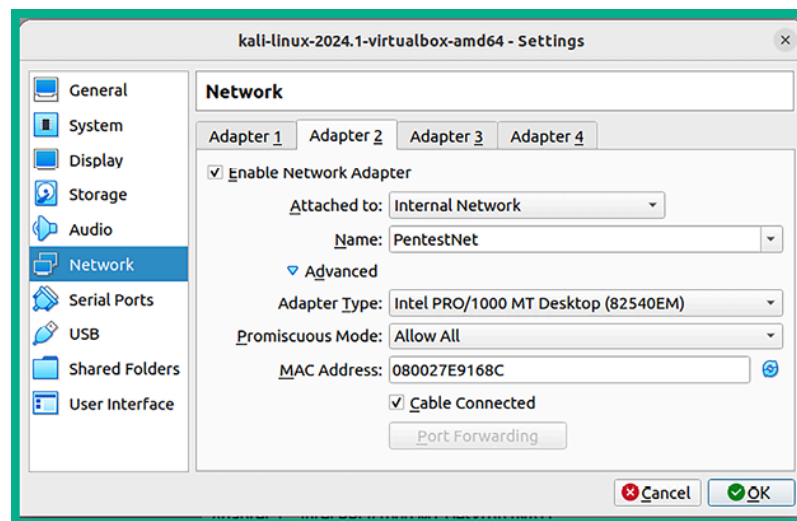


Figure 19.9: Network Adapter 2

10. Lastly, let's assign **Adapter 3** (vNIC 3) to the `RedTeamLab` network. Select the **Adapter 3** tab and use the following configurations:
1. Enable the network adapter.
  2. **Attached to: Internal Network.**
  3. **Name:** Manually enter `RedTeamLab` within the field.

#### 4. Promiscuous Mode: Allow All.

The following screenshot shows the preceding configurations applied to **Adapter 3** (vNIC 3):

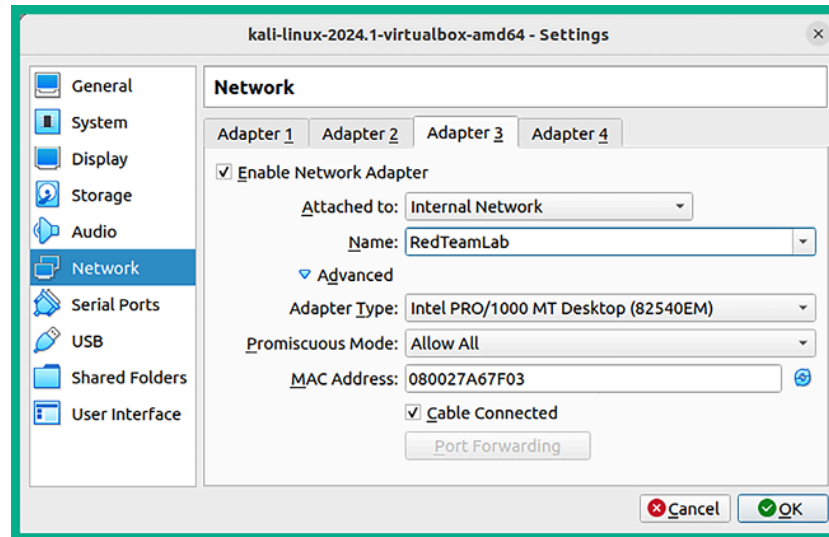


Figure 19.10: Network Adapter 3

After configuring the network settings on **Adapter 3**, disable it by un-checking **Enable Network Adapter** and click on **OK** to save the settings of the Kali Linux virtual machine. We will re-enable **Adapter 3** when it's needed in various chapters of this book.

## Setting up Metasploitable 3 on Ubuntu

In this section, you will learn how to build and deploy Metasploitable 3 (both the Windows Server and Linux server versions) on Ubuntu Desktop. The Windows Server version will be using a dual-homed network connection to both the **PentestNet** network ( `172.30.1.0/24` ) and the **HiddenNet** network ( `10.11.12.0/24` ). This setup will enable us to perform pivoting and lateral movement between different networks. Finally, the Linux server version will be connected to the **HiddenNet** network ( `10.11.12.0/24` ) only.

The following diagram shows the logical connections between systems and networks:

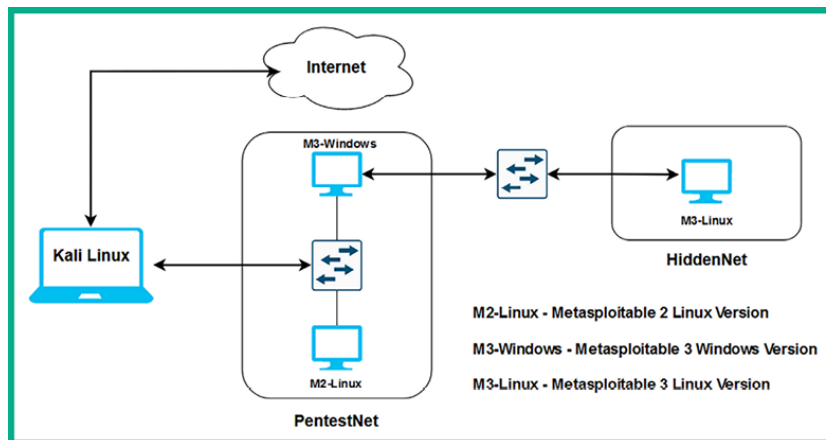


Figure 19.11: Low-level diagram

As shown in the preceding diagram, this topology goes into more depth on how the virtual machines are interconnected within our virtual lab environment. For instance, to access the Metasploitable 3 – Linux version, we will need to first compromise the Metasploitable 3 – Windows version via the **PentestNet** network, then pivot our attacks to the **HiddenNet** network.

## Part 1 – building the Windows Server version

To get started with building and deploying the Metasploitable 3 – Windows version, please use the following instructions:

1. Open Terminal on Ubuntu Desktop and use the following commands to install and set up Vagrant:

```

glen@ubuntu:~$ cd Downloads/
glen@ubuntu:~/Downloads$ wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/has
glen@ubuntu:~/Downloads$ echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.ha
glen@ubuntu:~/Downloads$ sudo apt update && sudo apt install vagrant

```

2. Next, use the following commands to reload and install additional plugins for Vagrant:

```

glen@ubuntu:~/Downloads$ vagrant plugin install vagrant-reload
glen@ubuntu:~/Downloads$ vagrant plugin install vagrant-vbguest

```

The following screenshot shows the execution of the preceding commands:

```
glen@ubuntu:~/Downloads$ vagrant plugin install vagrant-reload
Installing the 'vagrant-reload' plugin. This can take a few minutes...
Fetching vagrant-reload-0.0.1.gem
Installed the plugin 'vagrant-reload (0.0.1)!'
glen@ubuntu:~/Downloads$ vagrant plugin install vagrant-vbguest
Installing the 'vagrant-vbguest' plugin. This can take a few minutes...
Fetching micromachine-3.0.0.gem
Fetching vagrant-vbguest-0.32.0.gem
Installed the plugin 'vagrant-vbguest (0.32.0)!'
```

Figure 19.12: Reloading Vagrant plugins

- Next, use the following commands to load the Metasploitable 3 – Windows Server version onto your Ubuntu machine using Vagrant:

```
glen@ubuntu:~/Downloads$ vagrant box add rapid7/metasploitable3-win2k8
```

- Next, select option 1 to use VirtualBox as the preferred hypervisor, as shown below:

```
glen@ubuntu:~/Downloads$ vagrant box add rapid7/metasploitable3-win2k8
==> box: Loading metadata for box 'rapid7/metasploitable3-win2k8'
box: URL: https://vagrantcloud.com/api/v2/vagrant/rapid7/metasploitable3-win2k8
This box can work with multiple providers! The providers that it
can work with are listed below. Please review the list and choose
the provider you will be working with.

1) virtualbox
2) vmware
3) vmware_desktop

Enter your choice: 1
==> box: Adding box 'rapid7/metasploitable3-win2k8' (v0.1.0-weekly) for provider: virtualbox
box: Downloading: https://vagrantcloud.com/rapid7/boxes/metasploitable3-win2k8/versions/0.1.0-weekly/providers/virtualbox/unknown/vagrant.box
==> box: Successfully added box 'rapid7/metasploitable3-win2k8' (v0.1.0-weekly) for 'virtualbox'!
```

Figure 19.13: Reloading Vagrant plugins

- Once the download process is completed, use the following commands to rename the `rapid7-VAGRANTSLASH-metasploitable3-win2k8` folder:

```
glen@ubuntu:~/Downloads$ cd ~/.vagrant.d/boxes
glen@ubuntu:~/vagrant.d/boxes$ mv rapid7-VAGRANTSLASH-metasploitable3-win2k8 metasploitable3-win2k8
```

The following screenshot shows the successful execution of the preceding commands:

```

glen@ubuntu:~/Downloads$ cd ~/.vagrant.d/boxes
glen@ubuntu:~/vagrant.d/boxes$ ls -l
total 4
drwxrwxr-x 3 glen glen 4096 Apr 21 10:43 rapid7-VAGRANTSLASH-metasploitable3-win2k8
glen@ubuntu:~/vagrant.d/boxes$ mv rapid7-VAGRANTSLASH-metasploitable3-win2k8 metasploitable3-win2k8
glen@ubuntu:~/vagrant.d/boxes$ ls -l
total 4
drwxrwxr-x 3 glen glen 4096 Apr 21 10:43 metasploitable3-win2k8

```

Figure 19.14: Initializing the Metasploitable 3 image

- Next, use the following commands to start the build process of this virtual machine:

```

glen@ubuntu:~/vagrant.d/boxes$ vagrant init metasploitable3-win2k8
glen@ubuntu:~/vagrant.d/boxes$ vagrant up

```

The following screenshot shows the execution of the preceding commands:

```

glen@ubuntu:~/vagrant.d/boxes$ vagrant init metasploitable3-win2k8
A 'Vagrantfile' has been placed in this directory. You are now
ready to 'vagrant up' your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
'vagrantup.com' for more information on using Vagrant.
glen@ubuntu:~/vagrant.d/boxes$ vagrant up
B
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'metasploitable3-win2k8'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'metasploitable3-win2k8' version '0.1.0-weekly' is up to date...
==> default: Setting the name of the VM: boxes_default_1713711081673_83717
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
default: Adapter 1: nat
==> default: Forwarding ports...
default: 3389 (guest) => 3389 (host) (adapter 1)
default: 22 (guest) => 2222 (host) (adapter 1)
default: 5985 (guest) => 55985 (host) (adapter 1)
default: 5986 (guest) => 55986 (host) (adapter 1)
==> default: Running 'pre-boot' VM customizations...

```

Figure 19.15: Building a Metasploitable 3 VM

This process usually takes a few minutes to complete.

- After the process is completed, open VirtualBox. You will find a newly created virtual machine running as shown below:

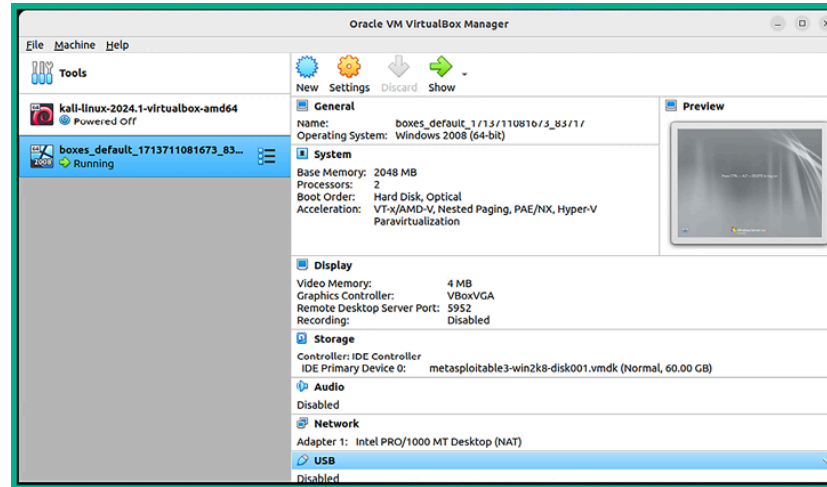


Figure 19.16: VirtualBox with the Metasploitable 3 VM

8. Select the Metasploitable 3 – Windows virtual machine and click on **Show to detect it from VirtualBox Manager**.
9. Once the virtual machine is detached, on the virtual machine menu bar, click on **Input | Keyboard | Insert Ctrl-Alt-Del**, as shown in the following screenshot:

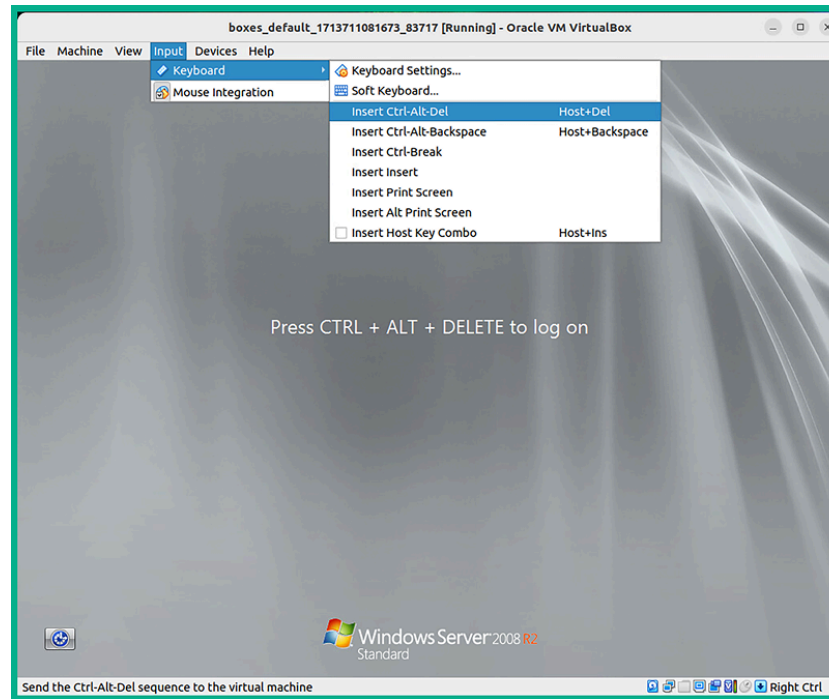


Figure 19.17: Input menu on VirtualBox

10. Select the `Administrator` account and use the default password, `vagrant`, to log in, as shown below:



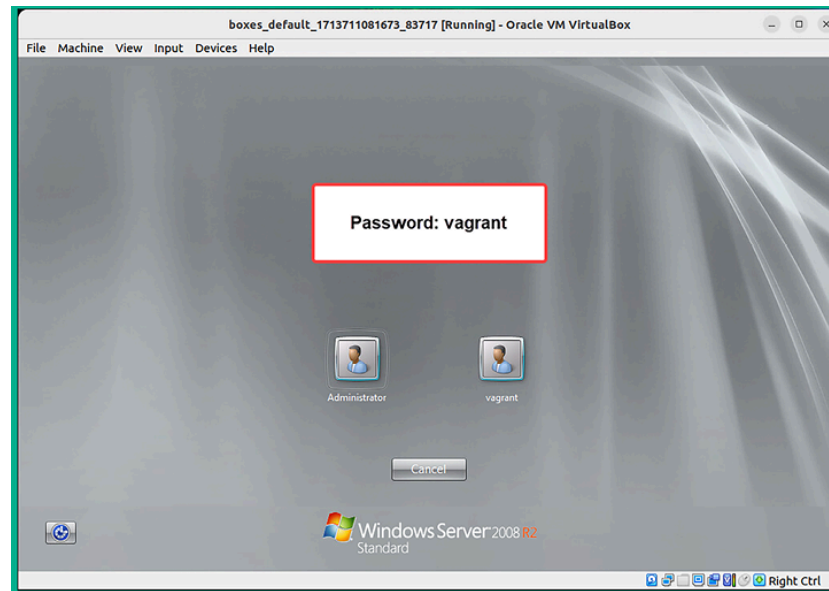


Figure 19.18: Login screen

11. Log in to the server and shut it down.
12. Once the Metasploitable 3 – Windows virtual machine is powered off, select the virtual machine and click on **Settings** as shown below:

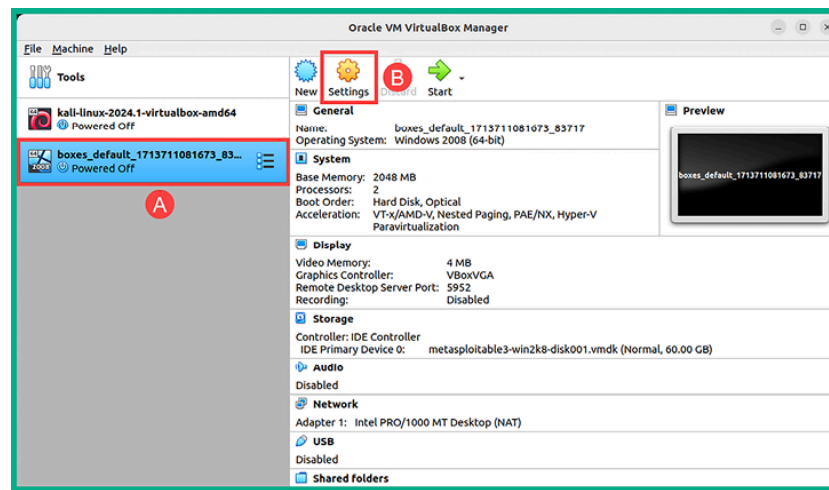


Figure 19.19: VirtualBox Manager

13. On the **General** category | **Basic** tab, change the default name of the virtual machine as shown below:

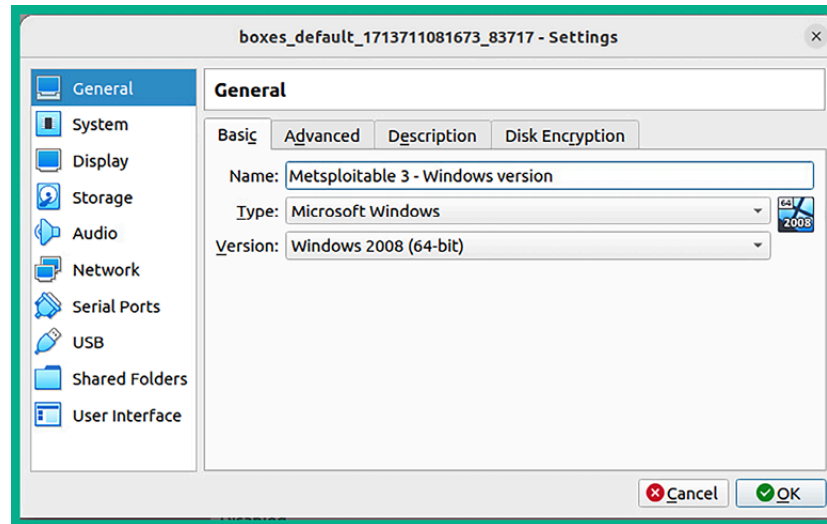


Figure 19.20: Virtual machine name

14. Next, select **Network** | **Adapter 1** and use the following configurations:

1. Enable the network adapter.
2. **Attached to: Internal Network.**
3. **Name:** Manually enter `PentestNet` within the field.
4. **Promiscuous Mode: Allow All.**

The following screenshot shows the preceding configurations applied to **Adapter 1**:

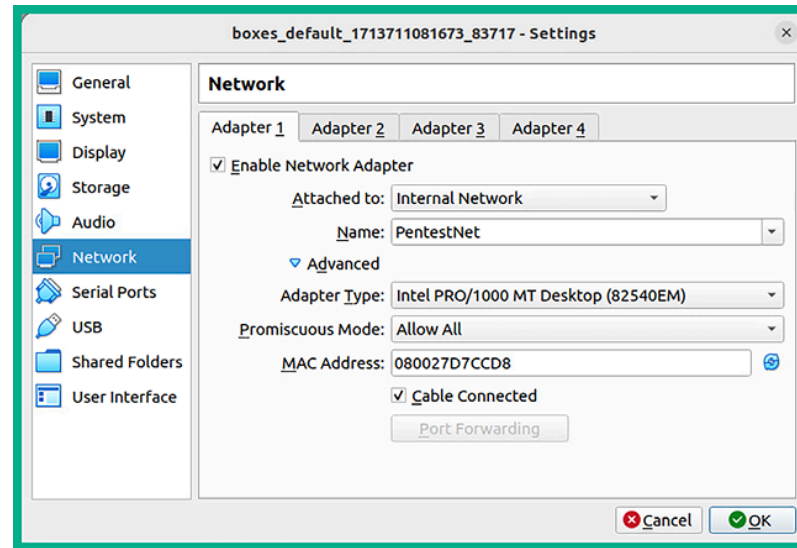


Figure 19.21: Network adapter 1

15. Next, select **Network** | **Adapter 2**, use the following configurations, and click on **Save**:

1. Enable the network adapter.
2. **Attached to: Internal Network.**
3. **Name:** Manually enter `HiddenNet` within the field.
4. **Promiscuous Mode: Allow All.**

The following screenshot shows the preceding configurations applied to **Adapter 2**:

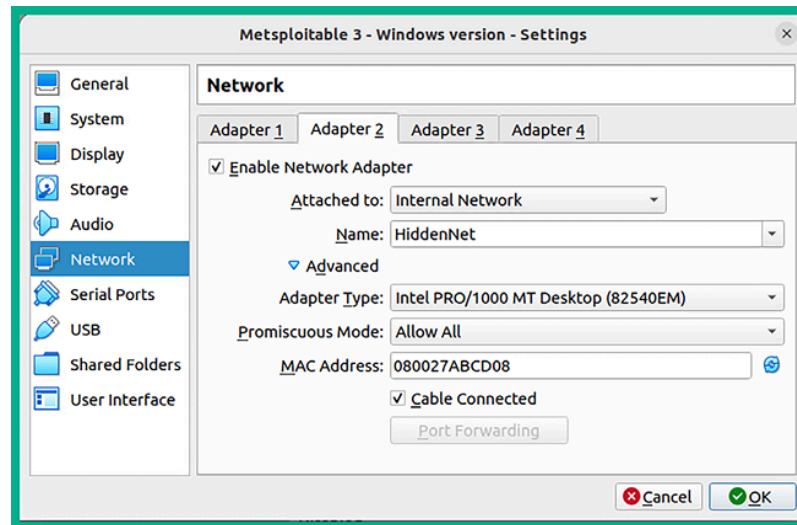


Figure 19.22: Network adapter 2

16. Next, power on the Metasploitable 3 – Windows virtual machine and log in using the Administrator account. When logged in, open the Windows Command Prompt and use the `ipconfig` command to verify this virtual machine is receiving IP addresses from the `172.30.1.0/24` and `10.11.12.0/24` networks, as shown below:

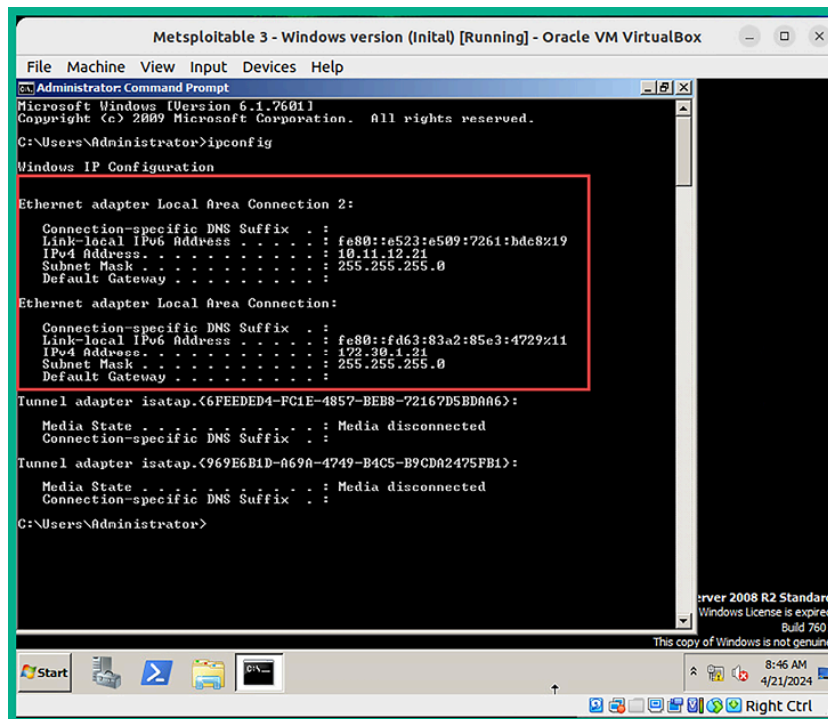


Figure 19.23: Network Adapters

17. Lastly, power off the virtual machine.

## Part 2 – building the Linux Server version

To get started with building and deploying Metasploitable 3 – Linux version, please use the following instructions:

1. On Ubuntu, open Terminal and use the following commands to download the Vagrant image for Metasploitable 3 – Linux version:

```
glen@ubuntu:~$ cd ~/.vagrant.d/boxes
glen@ubuntu:~/.vagrant.d/boxes$ vagrant box add rapid7/metasploitable3-ub1404
```

2. When you're prompted to choose a provider, select option 1, as shown below:

```

glen@ubuntu:~/.vagrant.d/boxes$ vagrant box add rapid7/metasploitable3-ub1404
==> box: Loading metadata for box 'rapid7/metasploitable3-ub1404'
box: URL: https://vagrantcloud.com/api/v2/vagrant/rapid7/metasploitable3-ub1404
This box can work with multiple providers! The providers that it
can work with are listed below. Please review the list and choose
the provider you will be working with.

1) virtualbox
2) vmware
3) vmware_desktop

Enter your choice:
Invalid choice. Try again: 1
==> box: Adding box 'rapid7/metasploitable3-ub1404' (v0.1.12-weekly) for provider: virtualbox
box: Downloading: https://vagrantcloud.com/rapid7/boxes/metasploitable3-ub1404/versions/0.1.12-weekl
y/providers/virtualbox/unknown/vagrant.box
==> box: Successfully added box 'rapid7/metasploitable3-ub1404' (v0.1.12-weekly) for 'virtualbox'!

```

Figure 19.24: Downloading Linux version

3. Next, delete the Vagrantfile by using the following commands:

```
glen@ubuntu:~/.vagrant.d/boxes$ rm Vagrantfile
```

The following screenshot shows the execution of the preceding commands:

```

glen@ubuntu:~/.vagrant.d/boxes$ rm Vagrantfile
glen@ubuntu:~/.vagrant.d/boxes$ ls -l
total 8
drwxrwxr-x 3 glen glen 4096 Apr 21 10:43 metasploitable3-win2k8
drwxrwxr-x 3 glen glen 4096 Apr 21 11:21 rapid7-VAGRANTSLASH-metasploitable3-ub1404

```

Figure 19.25: Removing the Vagrant file

4. Next, rename the `rapid7-VAGRANTSLASH-metasploitable3-ub1404` folder to `metasploitable3-ub1404` and start the initialization process for creating the virtual machine:

```

glen@ubuntu:~/.vagrant.d/boxes$ mv rapid7-VAGRANTSLASH-metasploitable3-ub1404 metasploitable3-ub1404
glen@ubuntu:~/.vagrant.d/boxes$ vagrant init metasploitable3-ub1404

```

The following screenshot shows the preceding commands executed successfully:

```

glen@ubuntu:~/.vagrant.d/boxes$ mv rapid7-VAGRANTSLASH-metasploitable3-ub1404 metasploitable3-ub1404
glen@ubuntu:~/.vagrant.d/boxes$ vagrant init metasploitable3-ub1404
A 'Vagrantfile' has been placed in this directory. You are now
ready to 'vagrant up' your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
'vagrantup.com' for more information on using Vagrant.
glen@ubuntu:~/.vagrant.d/boxes$

```

Figure 19.26: Starting the initialize process

5. Next, open File Explorer on Ubuntu Desktop and go to the `/home/<username>/ .vagrant.d/boxes/metasploitable3-ub1404/0.1.12-weekly/virtualbox` directory, then right-click on the **box.ovf** file and select **Open With Other Application:**

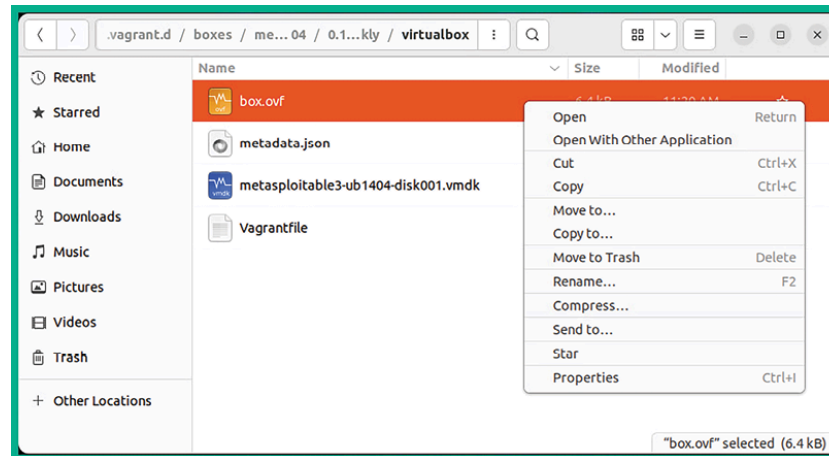


Figure 19.27: Virtual machine fines

6. On the **Select Application** window, click on **View All Application** and select **VirtualBox**. The following import window will appear. Click on **Import**:

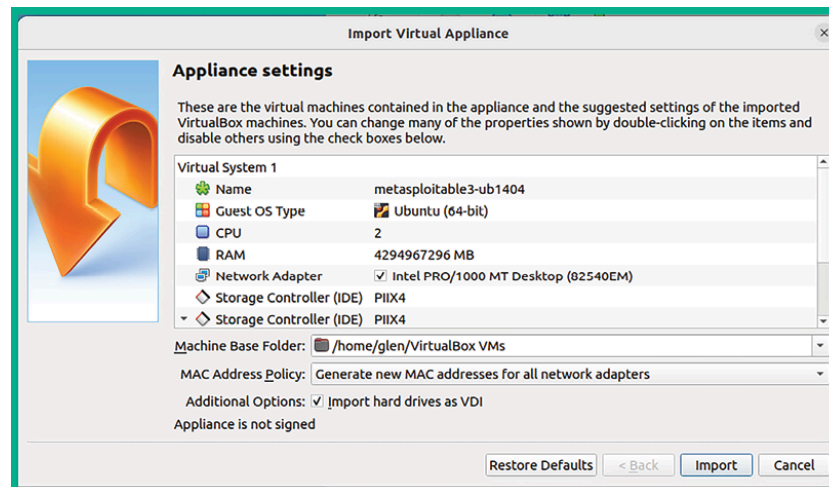


Figure 19.28: Virtual machine import window

- Once the import process is completed, the Metasploitable 3 – Linux virtual machine will appear in the VirtualBox manager. Select it and click on **Settings**, as shown below:

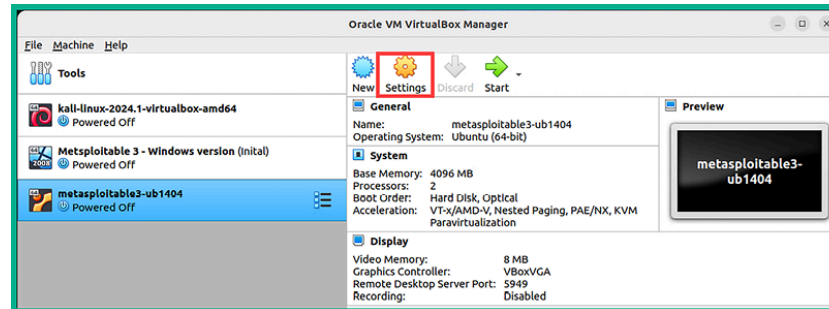


Figure 19.29: VirtualBox interface

- Next, select **Network | Adapter 1** and use the following configurations:
  - Enable the network adapter.
  - Attached to: Internal Network.**
  - Name:** Manually enter `HiddenNet` within the field.
  - Promiscuous Mode: Allow All.**

The following screenshot shows the preceding configurations applied to **Adapter 1**:



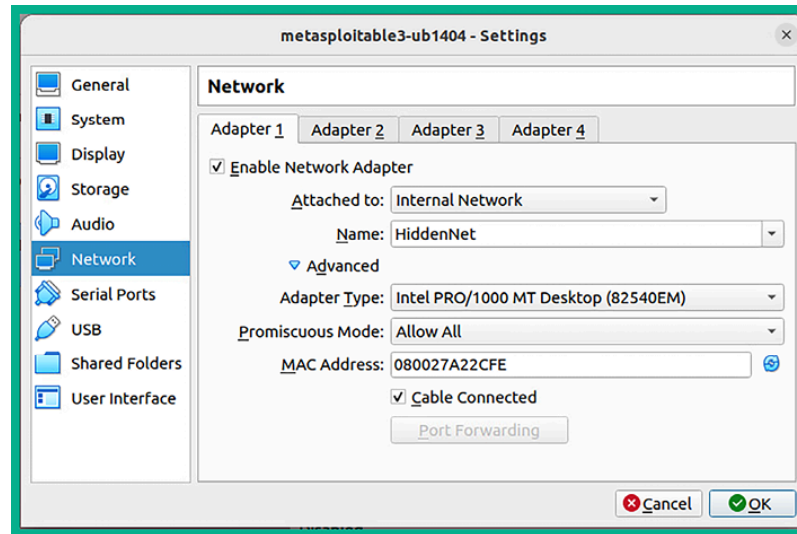


Figure 19.30: Network interface

9. Click on **OK** to save the settings.
10. Power on the Metasploitable 3 – Linux virtual machine and log in using the username `vagrant` and the password `vagrant` as shown below:

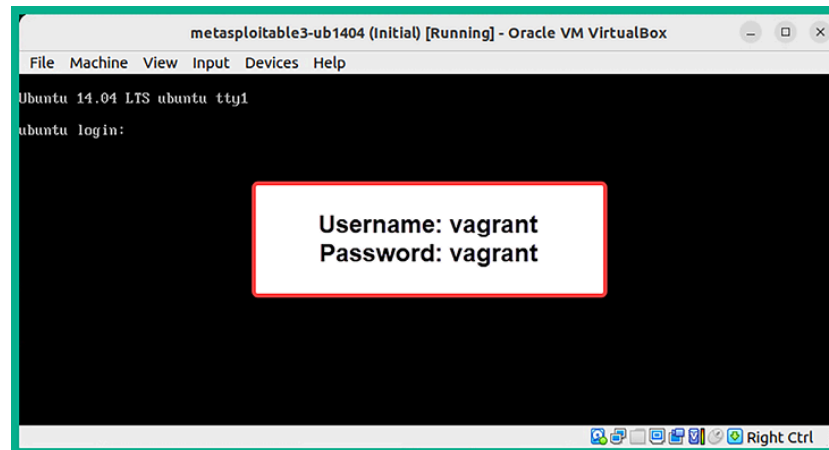


Figure 19.31: Metasploitable 3 interface

11. Next, use the `ip address` command to verify the virtual machine is receiving an IP address on the `10.11.12.0/24` network, as shown below:

*Figure 19.32: Metasploitable 3 network interface*

12. Lastly, you can use the `sudo halt` command to power off the virtual machine.

## Summary

This chapter covered how to set up a hypervisor, create virtual networks, and deploy Kali Linux and Metasploitable 3 in the lab environment. It's important to refer to *Chapter 2, Building a Penetration Testing Lab*, and *Chapter 3, Setting Up for Advanced Penetration Testing Techniques*, to continue building the lab.