2
Building a Penetration Testing Lab

As an aspiring ethical hacker and penetration tester, it's important to ensure that you do not disrupt or cause any sort of harm or damage to another person's systems or network infrastructure, such as that of your organization, when testing exploits and payloads or practicing your hacking skills. While there are many online tutorials, videos, and training materials you can read and view to gain knowledge, working in the field of penetration testing means continuously enhancing your offensive security skills. Many people can speak about hacking and explain the methodology quite clearly but don't know how to perform an attack. When learning about penetration testing, it's very important to understand the theory and how to use your skills to apply them to a simulated real-world cyberattack.

In this chapter, you will learn how to design and build a virtualized penetration testing lab environment on your personal computer and leverage virtualization technologies to reduce the cost and need of acquiring 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 vulnerable systems as your targets. It's important to always remember that when you are 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. To put it simply, hacking systems you do not own is illegal.

In this chapter, we will cover the following topics:

- Understanding the lab overview and technologies
- Setting up a hypervisor and virtual networks
- Setting up and working with Kali Linux
- Setting up a vulnerable web application
- Deploying Metasploitable 2 as a vulnerable machine
- Building and deploying Metasploitable 3

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/
- The **Open Web Application Security Project** (**OWASP**) Juice Shop – https://owasp.org/www-project-juice-shop/
- Metasploitable 2 (Linux) – https://sourceforge.net/projects/metasploitable/files/Metasploitable2/
- Metasploitable 3 (Windows and Linux) – https://app.vagrantup.com/rapid7

We'll be covering the process of setting up Kali Linux, Vagrant, the OWASP Juice Shop, and Metasploitable 2 and 3 in detail in the chapter.

Note

During the installation of Oracle VirtualBox, it's important to ensure the application is installed within the default location of your `C:` drive. In addition to that, please ensure you are using default settings during the installation process or else you may encounter issues.

Additionally, the technical setup of this lab is specifically designed to operate on Windows systems. Please be advised that this setup may not be compatible with Linux or macOS environments.

For Ubuntu users, please refer to the appendix chapter for instructions on setting up VirtualBox, creating virtual networks, deploying Kali Linux, and setting up Metasploitable 3 virtual machines on a Ubuntu Desktop operating system.

An overview of the lab setup and technologies used

Building a penetration testing lab enables you to create an environment that's safe for you to practice and enhance your offensive security skills, scale the environment to add new vulnerable systems and remove older legacy systems that you may no longer need, and even create additional virtual networks to pivot your attacks from one network to another.

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 based on their field of study. To some extent, this is true, but as technology advances, many downsides are associated with building a physical lab to practice your skills.

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 that utilize 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, which is 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 allow them to run 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, but 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 cyberattacks 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 **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 convenient for easily downloading additional applications, tools, and software packages.

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

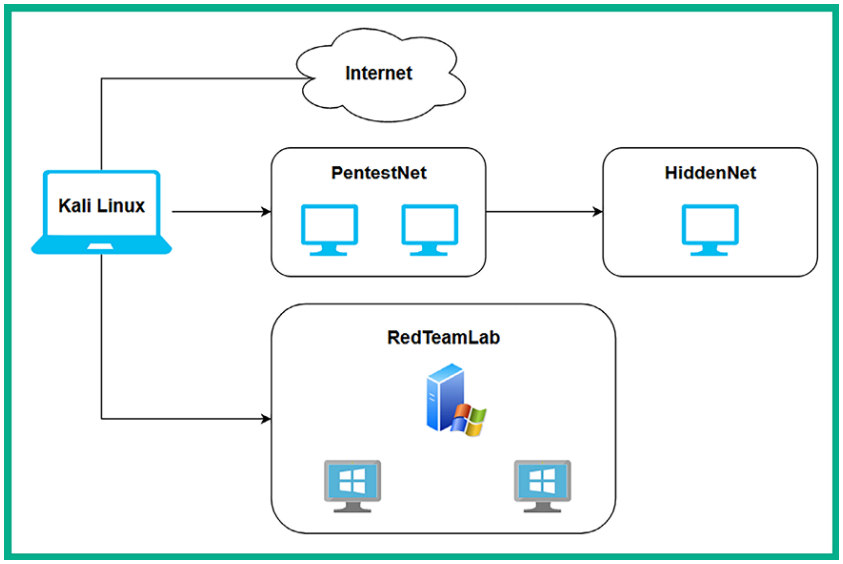


Figure 2.1: A high-level lab overview

As shown in the preceding diagram, there are four network zones, which are as follows:

- The internet for accessing online resources and is directly connected to the Kali Linux virtual machine.
- The **PentestNet** environment, which contains 2 vulnerable machines that are on the `172.30.1.0/24` network and is also directly connected to Kali Linux.
- The **RedTeamLab** environment that contains an **Active Directory** (**AD**) infrastructure with a Windows server and 2 clients that are on the `192.168.42.0/24` network, and it's 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 it's 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:

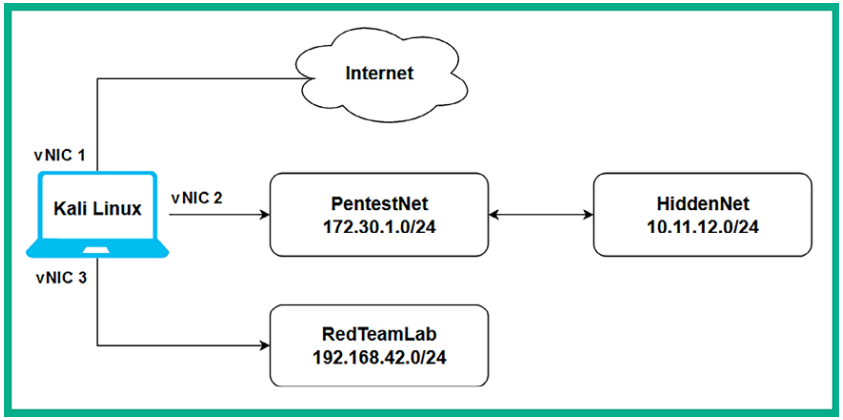


Figure 2.2: Technical level lab interview

As shown in the preceding diagram, the Kali Linux virtual machine will be assigned three network adapters, these 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 essential features of 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 **Basic Input/Output System (BIOS)** / **Unified Extensible Firmware Interface (UEFI)** firmware.

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

Let's get started!

Part 1 – setting up the hypervisor

As previously mentioned, there are many hypervisors in the industry, and we'll be using Oracle VM VirtualBox throughout this book. However, if you wish to use another hypervisor, ensure you configure it using the systems and network designs.

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

1. On your host computer, go to **https://www.virtualbox.org/wiki/Downloads** and choose the **Oracle VirtualBox Platform Package** that is suitable for your host operating system as shown in the following screenshot:

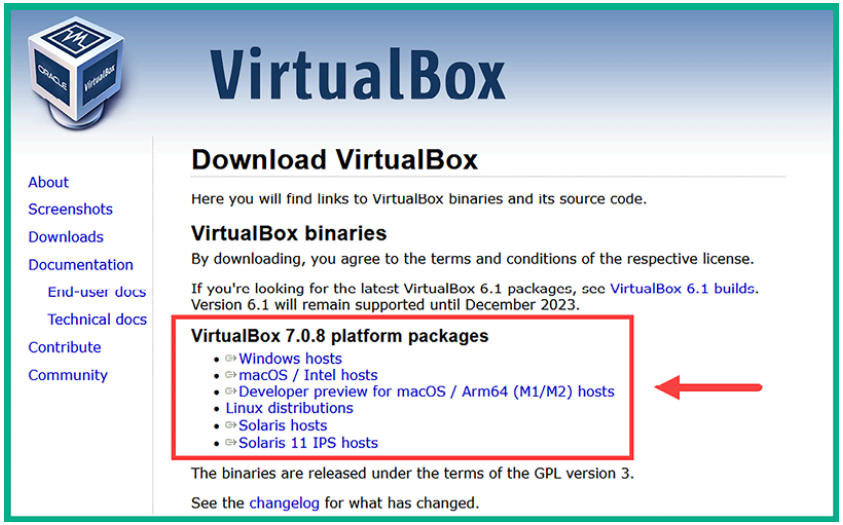


Figure 2.3: VirtualBox website

2. Next, you'll need to download the **Oracle VM VirtualBox Extension Pack** application. This enables additional functionality on the **VirtualBox Manager** application, such as creating virtually isolated networks on the host computer. On the same download page, scroll down a bit to find the download link as follows:

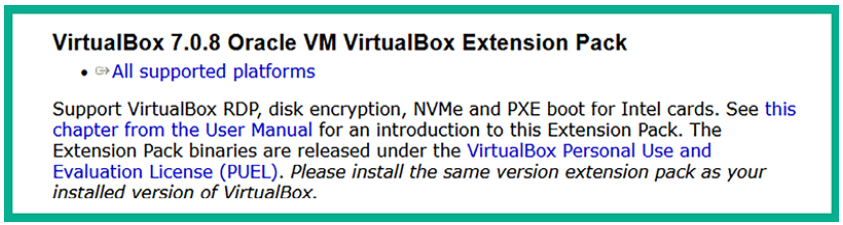


Figure 2.4: VirtualBox Extension Pack

3. Next, install the **Oracle VirtualBox Platform Package** that was downloaded during *step 1*. During the installation, use the default configurations. Once the application is installed on your host computer, the **VirtualBox Manager** interface will appear as shown in the following screenshot:

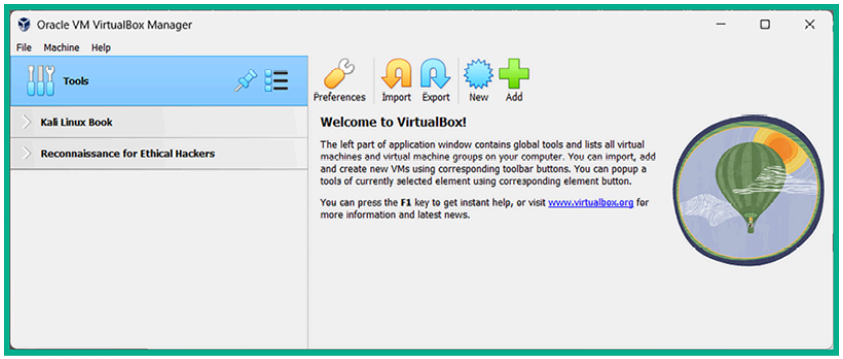


Figure 2.5: Oracle VirtualBox Manager

4. Next, close the **Oracle VM VirtualBox Manager** application as it's not needed at this time.
5. Next, to install the **Oracle VM VirtualBox Extension Pack**, simply right-click on the software package and choose **Open with** | **VirtualBox Manager**:

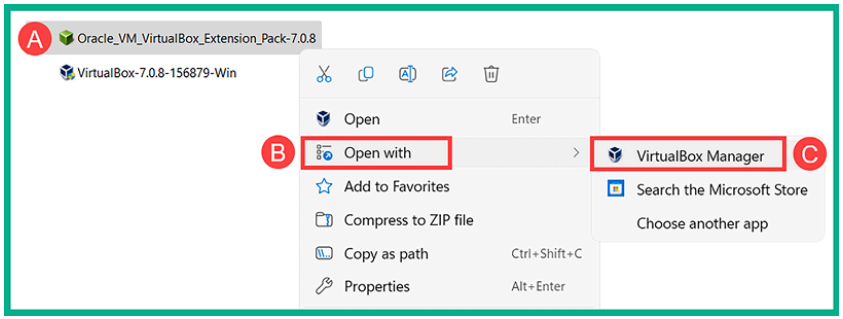


Figure 2.6: Opening with VirtualBox Manager

6. The **VirtualBox License** window will appear; ensure you read and click on **I Agree** to accept the agreement to proceed with its installation.

Once the installation is completed, you can close the **VirtualBox Manager** application until it's needed later.

Part 2 – creating virtually isolated networks

To get started setting up the virtually isolated networks, please follow these instructions:

1. Firstly, on your Windows host computer, open **Command Prompt**.
2. Next, use the following commands to change the present working directory to `C:\Program Files\Oracle\VirtualBox`:

```
C:\Users\Glen> cd C:\Program Files\Oracle\VirtualBox
```

3. Next, using the **vboxmanage** application, create a virtual **Dynamic Host Configuration Protocol** (**DHCP**) server for the virtual `PentestNet` network using the following commands:

```
C:\Program Files\Oracle\VirtualBox> vboxmanage dhcpserver add --network=PentestNet --server-ip=172.30.1.1 --lower-ip=172
```

The following snippet shows the preceding commands executed in the **Command Prompt**:

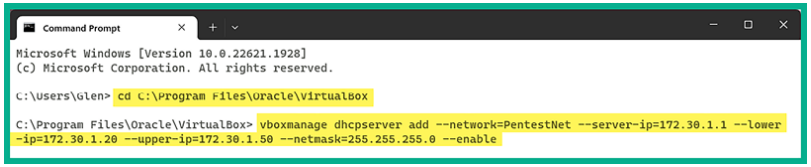


Figure 2.7: Creating the first virtual network

Upon executing the preceding commands, the **vboxmanage** application creates a DHCP server that will automatically assign an IP address within the range from `172.30.1.1` – `172.30.1.254` to any systems that are connected to the `PentestNet` network on the hypervisor.

Note

You can use the `vboxmanage list dhcpservers` command to view all DHCP servers and their configurations that are enabled on your host computer via VirtualBox.

4. Next, use the following commands to create a new DHCP server for the `HiddenNet` network:

```
C:\Program Files\Oracle\VirtualBox> vboxmanage dhcpserver add --network=HiddenNet --server-ip=10.11.12.1 --lower-ip=10.1
```

The following snippet shows the execution of the preceding commands:

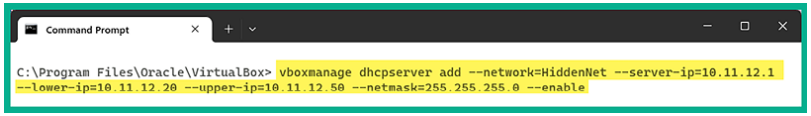


Figure 2.8: Creating the second virtual network

When the preceding commands are executed, it will create another virtual DHCP server that will automatically assign IP addresses within the range of `10.11.12.1` – `10.11.12.20` to any virtual machines that are connected to the `HiddenNet` network.

5. Next, create another DHCP server and virtual network that will be assigned to the `RedTeamLab` network by using the following commands:

```
C:\Program Files\Oracle\VirtualBox> vboxmanage dhcpserver add --network=RedTeamLab --server-ip=192.168.42.1 --lower-ip=1
```

The following snippet shows the execution of the preceding commands to create another virtual DHCP server:

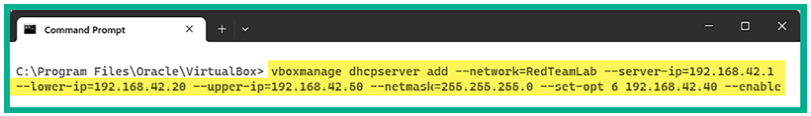


Figure 2.9: Creating the third virtual network

Unlike the previous steps, the commands used to create the `RedTeamLab` network were modified to specify a **Domain Name System (DNS)** server address to virtual machines that are connecting to this virtual network. The DNS server address will be useful when setting up the AD lab environment.

At this point, both the hypervisor and virtual networks are configured. Next, you will learn how to deploy and set up Kali Linux as a virtual machine within our lab environment.

Setting up and working with Kali Linux

Kali Linux is one of the most popular Linux distributions within the cybersecurity industry as it contains over 300 pre-installed software packages that are designed for mostly offensive security assessments. Kali Linux is built on the Debian flavor of Linux and, being a free operating system, it has gained a lot of attention over the years by cybersecurity professionals in the industry. It has a lot of features and tools that make a penetration tester's or security engineer's job a bit easier when they're working.

Ethical hackers and penetration testers commonly use Kali Linux to perform passive reconnaissance (covered in *Chapters 4* and *5*), scanning and enumeration (covered in *Chapter 6*), exploitation (covered in *Chapter 8*), and even post-exploitation techniques (covered in *Chapters 10* and *11*) on targeted systems and networks. While many folks usually think Kali Linux is designed only for offensive security professionals such as penetration testers, it's commonly used by system administrators and even network security professionals within the technology industry to test their security controls and systems for security vulnerabilities.

In this section, you will learn how to set up Kali Linux as a virtual machine, establish network connectivity to the internet and to our virtually isolated networks, and learn the basics of Kali Linux. Let's get started!

Part 1 – deploying Kali Linux as a virtual machine

There are many types of deployment models for Kali Linux, from performing a bare-metal installation directly on hardware to installing it on Android devices. To keep our lab setup process simple and easy to follow, you will learn how to set up Kali Linux as a virtual machine within the Oracle VM VirtualBox application. This method ensures you can be up and running very quickly.

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

1. Firstly, go to the official Kali Linux website at **https://www.kali.org/get-kali/** and click on **Virtual Machines** as shown here:

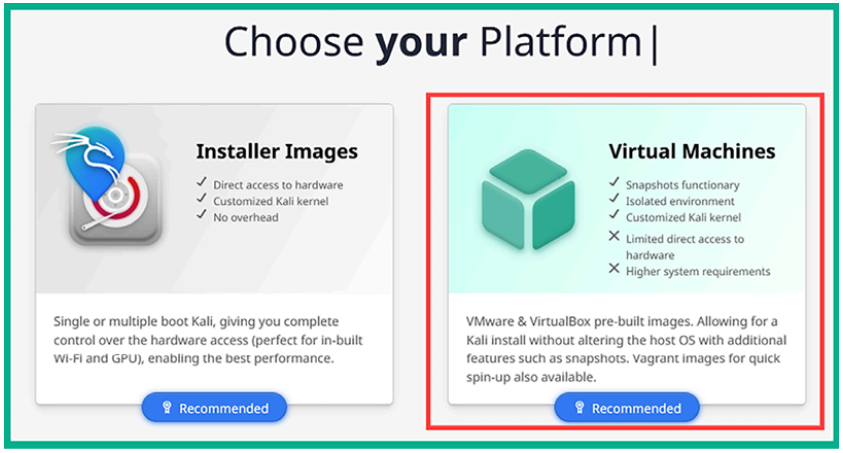


Figure 2.10: Kali Linux website

2. Next, click on **VirtualBox 64** to download the VirtualBox image of Kali Linux 2023:

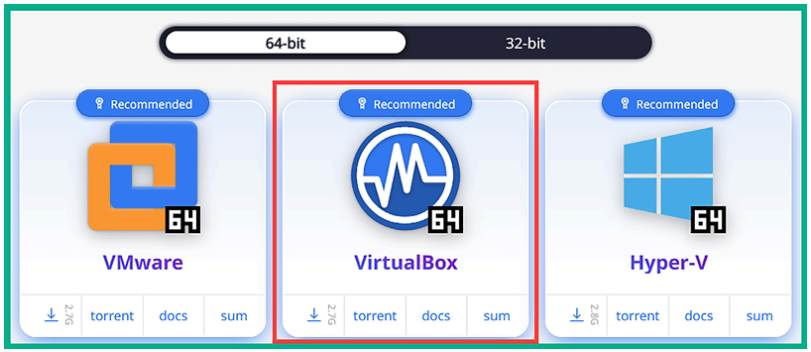


Figure 2.11: Kali Linux download section

The download file is a compressed folder with the `.7z` extension.

3. Next, to extract the contents from the compressed folder; you will need to download and install the 7-**Zip** application from **https://www.7-zip.org/download.html**.
4. Next, open the 7-**Zip File Manager** application, navigate to the directory with the Kali Linux compressed folder, select the file, and click on **Extract**:

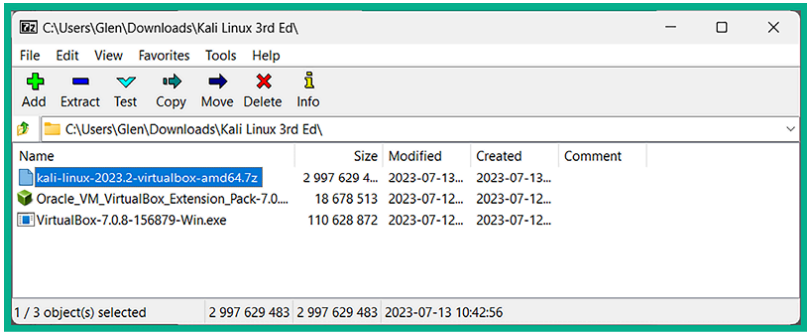


Figure 2.12: 7-Zip application

Next, the file extraction window will appear – click on **OK** to proceed:

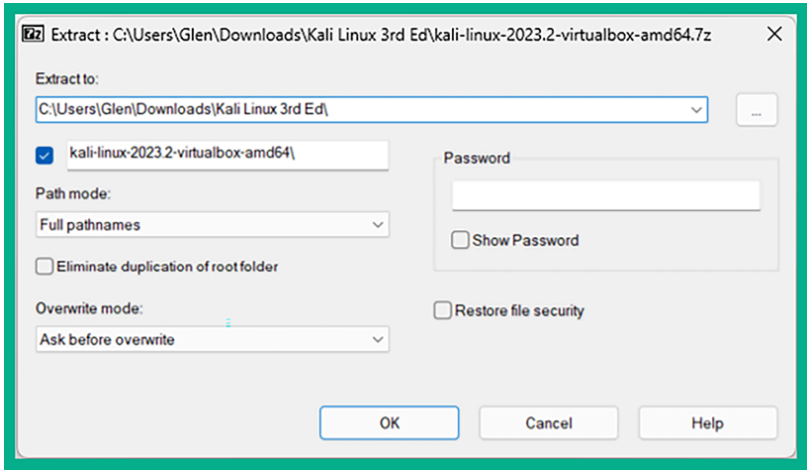


Figure 2.13: Extracting Kali Linux virtual files

The extraction process will begin and take a few seconds or minutes to complete. After the extraction is completed, you will see a new folder within the **7-Zip File Manager** application. This means the contents were successfully extracted and you can now close the application.

5. Next, open **Windows Explorer** and go to the directory that has the extracted contents. There you will see two files – right-click on the **VirtualBox Machine Definition** file and select **Open with** | **VirtualBox Manager**, as shown here:

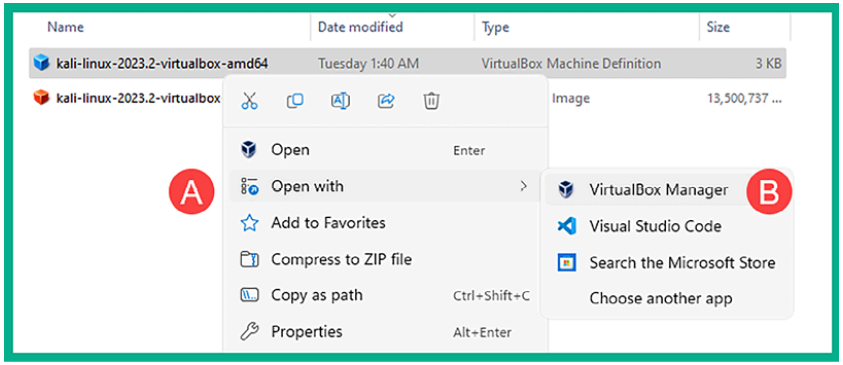


Figure 2.14: Opening Kali Linux with VirtualBox Manager

6. The **Oracle VM VirtualBox Manager** application will automatically open and import the Kali Linux virtual machine, as shown here:

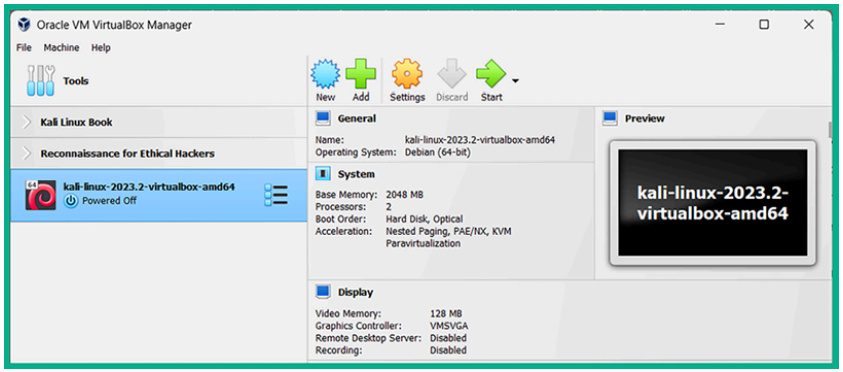


Figure 2.15: VirtualBox with Kali Linux

Before powering on the Kali Linux virtual machine, there are a few customizations that need to be made to the virtual machine settings.

Part 2 – customizing Kali Linux and its network adapters

The following instructions will guide you in customizing the Kali Linux virtual machine environment and ensuring it's aligned with our virtualized penetration testing lab topology. In addition, you will learn how to attach each vNIC (network adapter) to the internet, `PentestNet`, and `RedTeamLab` virtual networks.

To get started customizing the Kali Linux virtual environment, please follow these instructions:

1. Firstly, ensure the **Nested VT-x/AMD-V** virtualization feature is accessible between the virtual machine and the processor on your computer – we will need to execute the following commands within the Windows **Command Prompt**:

```
C:\Users\Glen> cd C:\Program Files\Oracle\VirtualBox
C:\Program Files\Oracle\VirtualBox> VBoxManage.exe list vms
```

NOTE

The `VBoxManage.exe list vms` command enables us to view a list of all the virtual machines, as well as their names and IDs within Oracle VM VirtualBox Manager.

2. Next, using the name of the newly imported Kali Linux virtual machine, use the following commands to enable the **Nested VT-x/AMD-V** feature on the virtual machine:

```
C:\Program Files\Oracle\VirtualBox> VBoxManage.exe modifyvm "kali-linux-2023.2-virtualbox-amd64" --nested-hw-virt on
```

Ensure you substitute the name of your Kali Linux virtual machine (shown in *step 1*) with the name displayed within the quotation marks, as shown here:

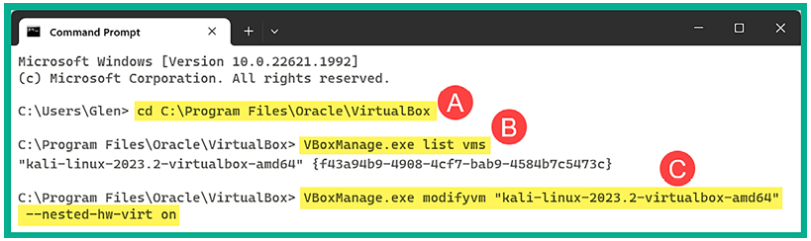


Figure 2.16: Enabling nested virtualization

3. Next, in Oracle **VM VirtualBox Manager**, select the **Kali Linux virtual machine** and click on **Settings** as shown here:

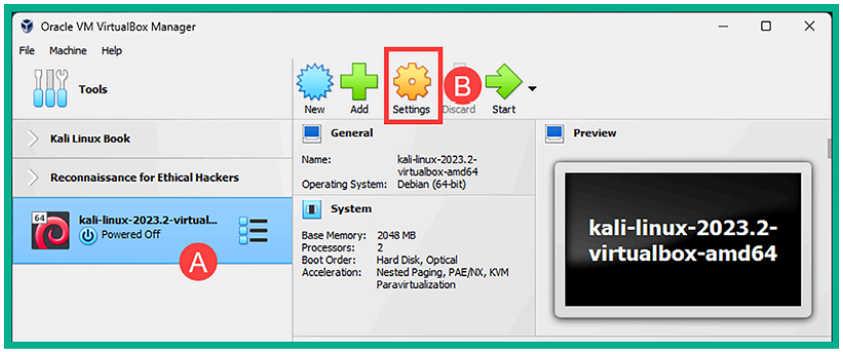


Figure 2.17: Settings icon

4. To adjust the amount of memory (RAM) allocated to this virtual machine, go to **System** | **Motherboard** | **Base Memory**, as shown here:

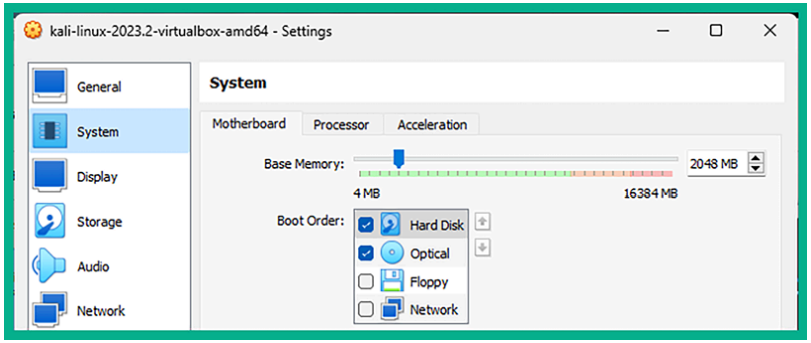


Figure 2.18: Kali Linux settings menu

It's recommended to never assign memory within the yellow and red zones of the Base Memory scale. Kali Linux can run efficiently on 2 GB of memory; however, if your system has more than 8 GB available, then consider allocating 4 GB to the Kali Linux virtual machine to ensure password-cracking tools such as hashcat can run smoothly during later chapters.

Additionally, within the **System** | **Processor** tab, you can modify the number of virtual CPU cores that are allocated to this virtual machine. Using between one and two cores is sufficient; however, you can assign more depending on the available hardware resources on your computer.

5. Next, let's connect the **Kali Linux virtual machine** to your physical network to access the internet. 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 your 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):

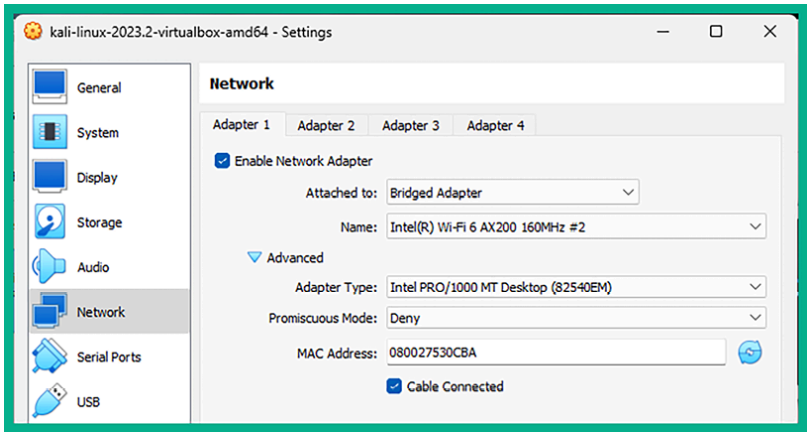


Figure 2.19: Network | Adapter 1

6. 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**

Note

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

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

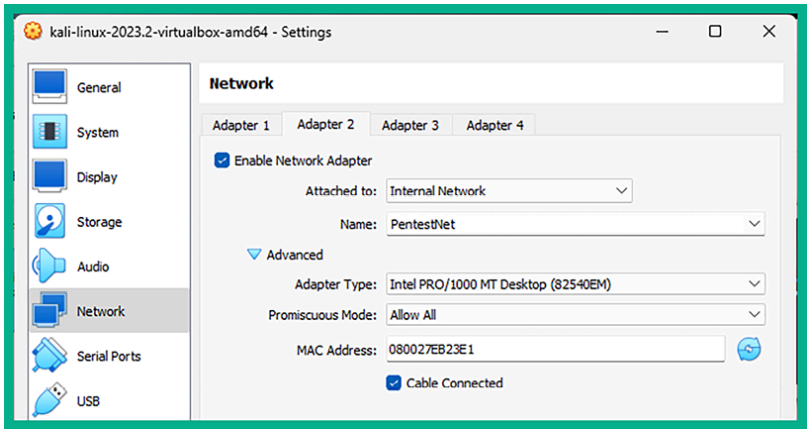


Figure 2.20: Network | Adapter 2

7. 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):

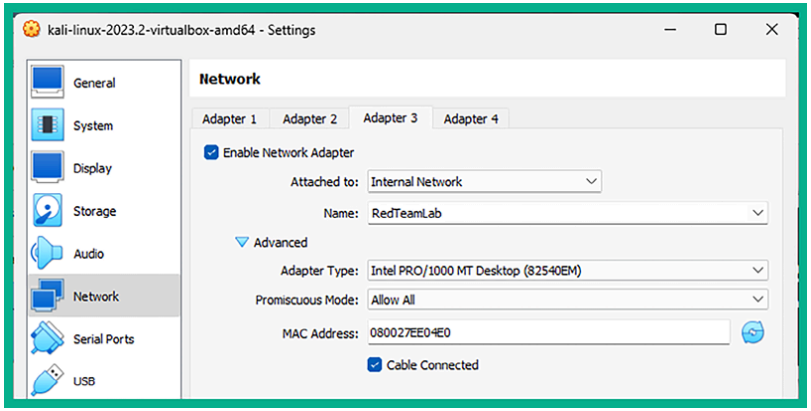


Figure 2.21: Network | Adapter 3

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

At this point, we have configured all three virtual network adapters on the Kali Linux virtual machine. One adapter provides connectivity to the internet via the physical adapter on your host computer, and the other two virtual adapters are connected to the virtual networks (`PentestNet` and `RedTeamLab`).

Part 3 – getting started with Kali Linux

Many first-time users are always excited to log in to their first attacker machine, especially a machine that's designed to help ethical hackers and penetration testers discover and exploit security vulnerabilities on targeted systems and networks.

The following instructions will help you get started with Kali Linux:

1. Firstly, open **Oracle VM VirtualBox Manager**, select the **Kali Linux virtual machine**, and click on **Start** to power on.
2. Next, a log-in prompt will appear; use the username: `kali` and password: `kali` to log in to the desktop:

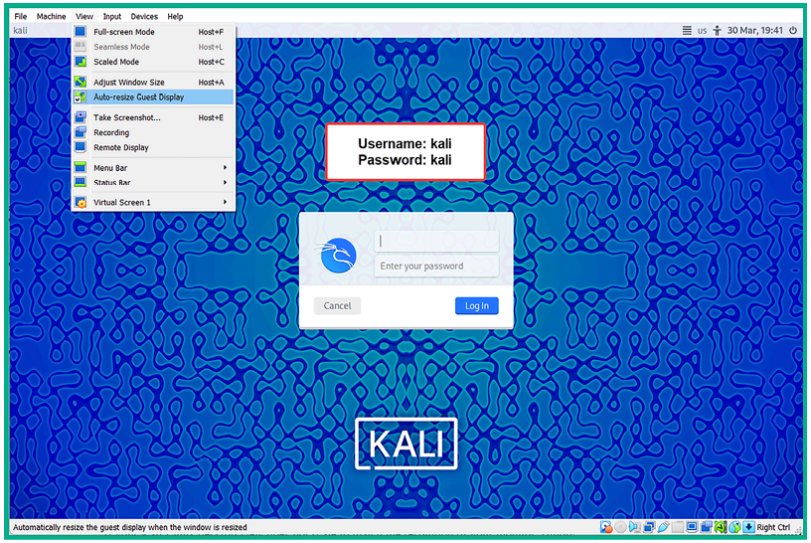
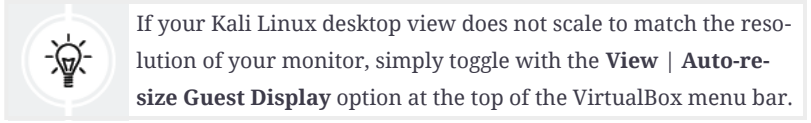


Figure 2.22: The Kali Linux Log In screen



If your Kali Linux desktop view does not scale to match the resolution of your monitor, simply toggle with the **View** | **Auto-resize Guest Display** option at the top of the VirtualBox menu bar.

3. Once you've logged in to the Kali Linux operating system, to view a list of available tools, click on the Kali Linux icon in the top-left corner of the desktop, as shown here:

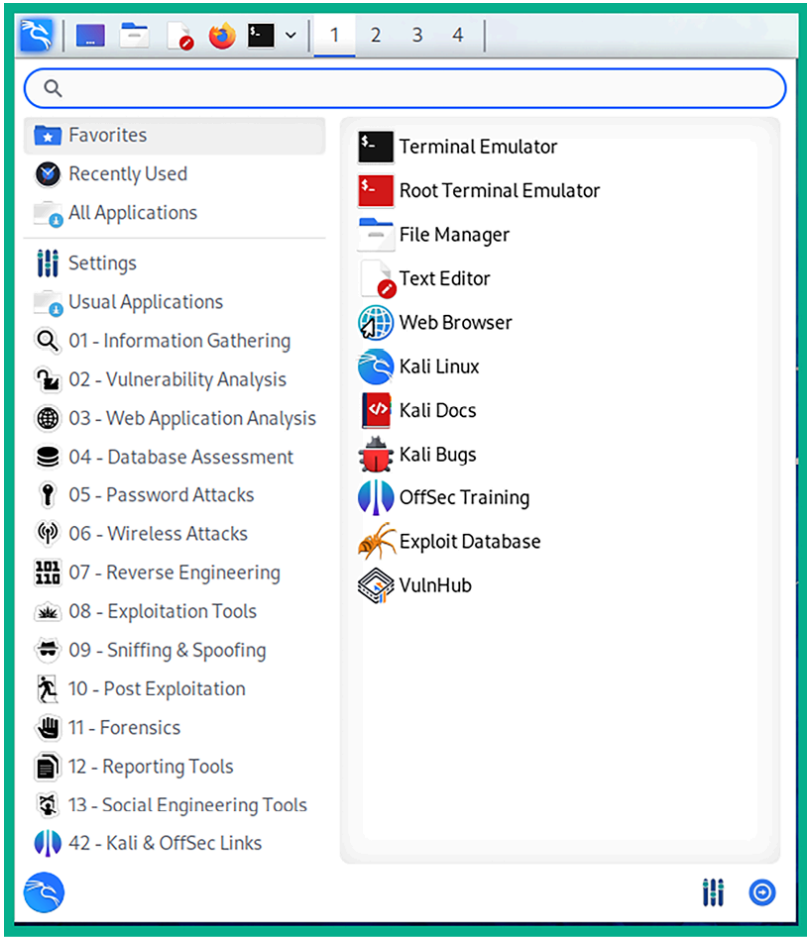


Figure 2.23: The Kali Linux application menu

As shown in the preceding screenshot, the pre-installed tools are all categorized based on the sequential order of performing ethical hacking and penetration testing exercises. For instance, all the tools that are commonly used for reconnaissance can be easily found within the **01 – Information Gathering** category, while wireless penetration testing tools are found within the **06 – Wireless Attacks** category.

Note

Throughout this book, you will mostly be working with the Linux terminal and learning many commands along the way. Don't worry if this is your first time working with Linux and com-

mands; it will be a new learning experience and fun to work with new technologies and develop your offensive security skills to simulate real-world cyberattacks.

4. Sometimes, Kali Linux does not communicate properly to the internet when its internet-facing interface is assigned both an IPv4 and IPv6 address. To disable IPv6 on Kali Linux, click on the Kali Linux icon in the top-left corner and select the **Settings Manager** icon, as shown here:

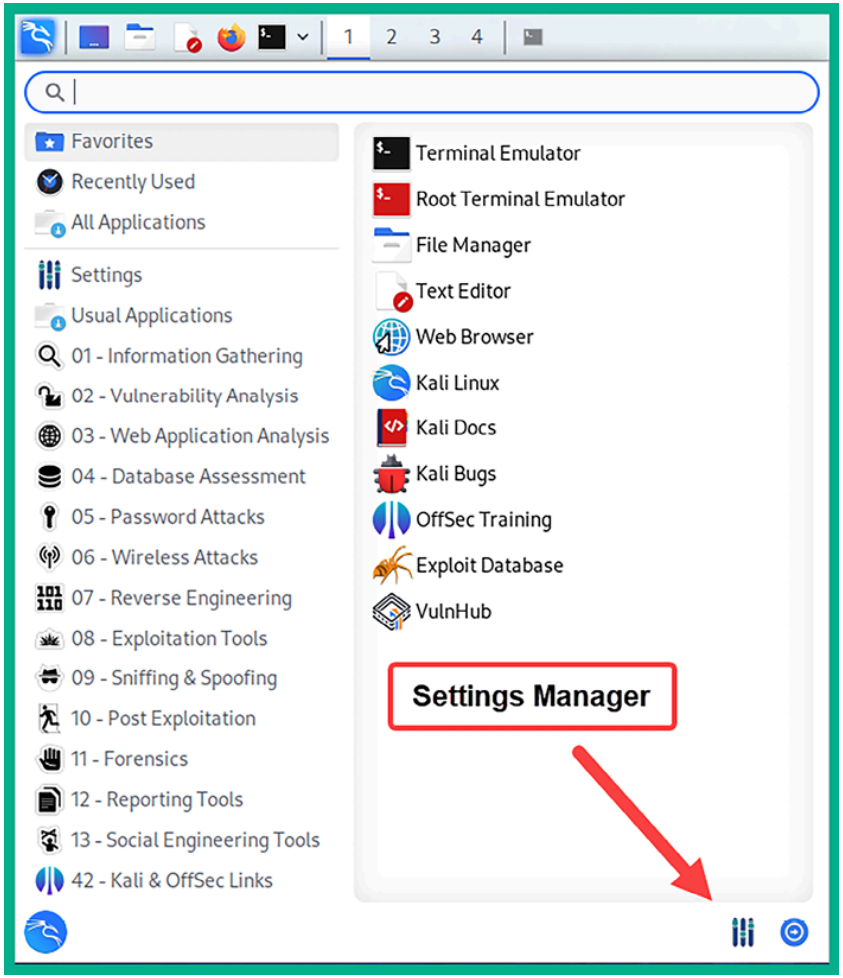


Figure 2.24: Locating the Settings Manager

5. The **Settings** window will appear. Here, click on **Advanced Network Configuration**, as shown in the following screenshot:

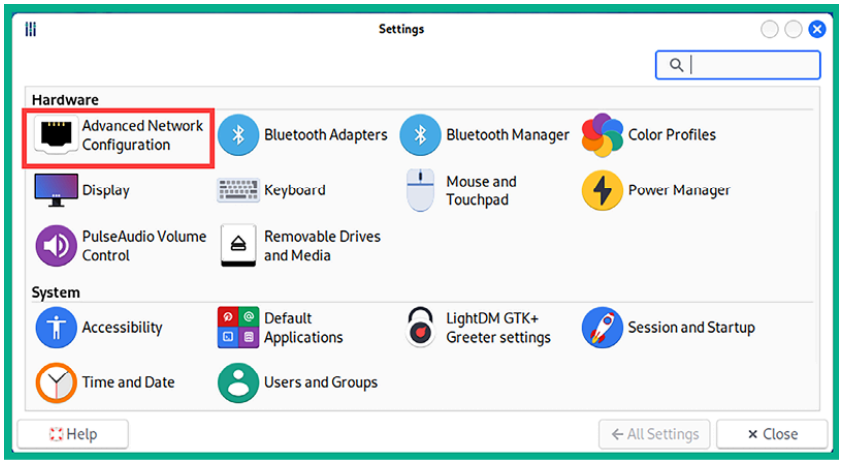


Figure 2.25: Locating the network settings

6. Next, the **Network Connections** window will appear. Here, select **Wired connection 1** (vNIC 1) and click on the gear icon, as shown in the following screenshot:

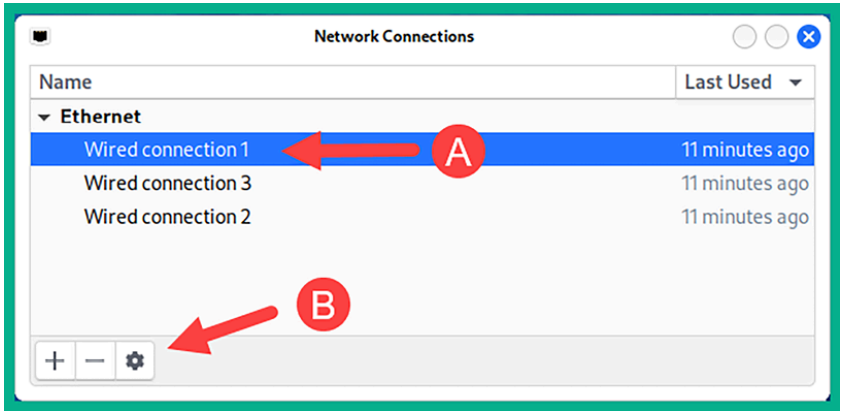


Figure 2.26: Selecting Wired connection 1

7. Next, the **Editing Wired connection 1** window appears. Here, select the **IPv6 Settings** tab, change **Method** to **Disabled**, and click on **Save**, as shown in the

following screenshot:

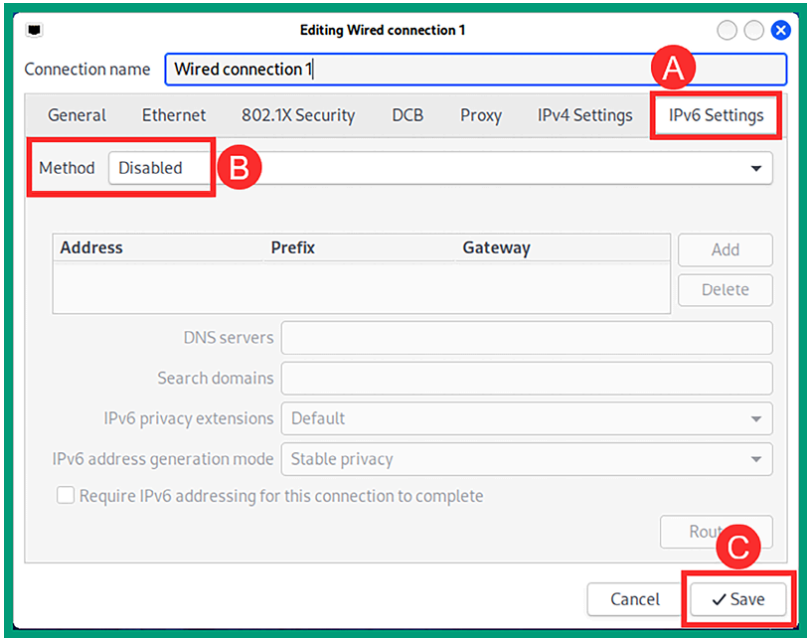


Figure 2.27: Disabling IPv6

You can now close the **Network Connections** window and the **Settings** menu.

8. Next, let's determine whether our Kali Linux virtual machine is receiving an IP address on each of its network adapters that are connected to the internet, `PentestNet`, and `RedTeamLab` networks. To open the **Linux terminal**, click on the Kali Linux icon on the top-left corner and select **Terminal Emulator**, then execute the `ip address` command shown here:

```
kali@kali:~$ ip address
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 fq_codel state UP group default qlen 1000
    link/ether 08:00:27:53:0c:ba brd ff:ff:ff:ff:ff:ff
    inet 172.16.17.15/24 brd 172.16.17.255 scope global dynamic noprefixroute eth0
       valid_lft 86152sec preferred_lft 86152sec
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:eb:23:e1 brd ff:ff:ff:ff:ff:ff
    inet 172.30.1.50/24 brd 172.30.1.255 scope global dynamic noprefixroute eth1
       valid_lft 353sec preferred_lft 353sec
    inet6 fe80::c280:130d:eca4:e07c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ee:04:e0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.42.27/24 brd 192.168.42.255 scope global dynamic noprefixroute eth2
       valid_lft 355sec preferred_lft 355sec
    inet6 fe80::362:d183:77b6:23d8/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Figure 2.28: Viewing all network adapters

As shown in the preceding screenshot, there are four network adapters on the Kali Linux virtual machine:

1. **lo**: This is the loopback network adapter, which enables the operating system to communicate with self-hosted applications and vice versa.
2. **eth0**: This network adapter is vNIC 1, based on our lab topology diagram, and it's represented as Network Adapter 1 on the virtual machine setting shown on VirtualBox Manager that's connected to the internet via the physical network. The **inet** address is the IP address that's allocated to the interface.
3. **eth1**: This is vNIC 2, according to the lab topology diagram, and it is Network Adapter 2, as shown on the VirtualBox Manager within the virtual machine setting that's connected to the `PentestNet` network (`172.30.1.0/24`) environment.
4. **eth3**: This is vNIC 3, according to the lab topology diagram, and it is Network Adapter 3, as shown on the VirtualBox Manager within the virtual machine setting that's connected to the `RedTeamLab` network (`192.168.42.0/24`) environment.

9. Next, let's check the internet connectivity and determine whether DNS resolution is working properly on our Kali Linux virtual machine. In the **Terminal**, use the following command to send four **Internet Control Message Protocol (ICMP)** messages to **www.google.com**:

```
kali@kali:~$ ping www.google.com -c 4
```

The following screenshot shows that the Kali Linux operating system was able to resolve the hostname to an IP address and successfully reach Google's web server on the internet:

```
kali@kali:~$ ping www.google.com -c 4
PING www.google.com (192.178.50.68) 56(84) bytes of data.
64 bytes from tzmiaa-ad-in-f4.1e100.net (192.178.50.68): icmp_seq=1 ttl=109 time=47.8 ms
64 bytes from tzmiaa-ad-in-f4.1e100.net (192.178.50.68): icmp_seq=2 ttl=109 time=48.7 ms
64 bytes from tzmiaa-ad-in-f4.1e100.net (192.178.50.68): icmp_seq=3 ttl=109 time=48.5 ms
64 bytes from tzmiaa-ad-in-f4.1e100.net (192.178.50.68): icmp_seq=4 ttl=109 time=48.4 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 47.845/48.370/48.708/0.319 ms
```

Figure 2.29: Checking internet connectivity

10. Finally, to change the default password for the username: `kali`, use the `passwd` command shown in the following screenshot:

```
kali@kali:~$ passwd
Changing password for kali.
Current password:
New password:
Retype new password:
passwd: password updated successfully
```

Figure 2.30: Changing the default password

Note

While entering passwords on the Linux terminal, they are invisible for security reasons.

Part 4 – updating repository sources and packages

At times, a tool may not be working as expected or even crash unexpectedly on us during a penetration test or security audit. Developers often release updates for their applications and software packages. These updates are intended to fix bugs and add new features to the user experience.

Let's learn how to update sources and packages by following these steps:

1. To update the local package repository list on Kali Linux, use the `sudo apt update` command shown here:

```
kali@kali:~$ sudo apt update
[sudo] password for kali:
Get:1 http://mirrors.jevincanders.net/kali kali-rolling InRelease [41.2 kB]
Get:2 http://mirrors.jevincanders.net/kali kali-rolling/main amd64 Packages [19.4 MB]
Get:3 http://mirrors.jevincanders.net/kali kali-rolling/main amd64 Contents (deb) [45.4 MB]
Get:4 http://mirrors.jevincanders.net/kali kali-rolling/contrib amd64 Packages [115 kB]
Get:5 http://mirrors.jevincanders.net/kali kali-rolling/contrib amd64 Contents (deb) [164 kB]
Get:6 http://mirrors.jevincanders.net/kali kali-rolling/non-free amd64 Packages [217 kB]
Get:7 http://mirrors.jevincanders.net/kali kali-rolling/non-free amd64 Contents (deb) [918 kB]
Fetched 66.3 MB in 15s (4,476 kB/s)
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
554 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Figure 2.31: Updating software packages list

2. By updating the package repository list on your Kali Linux machine, when you use the `sudo apt install <package-name>` command to install a new software package, Kali Linux will retrieve the latest version of the application and update it from the official sources.

The `source.list` file does not always update properly. To ensure you have the right settings on your Kali Linux machine, please see the official documentation on Kali Linux repositories at **https://www.kali.org/docs/general-use/kali-linux-sources-list-repositories/**.

Upgrading the software packages on Kali Linux usually introduces security updates and fixes known issues but also creates new issues. For instance, I've encountered that, after a software upgrade is completed on Kali Linux running VirtualBox, all network adapters are unable to obtain an IP address. Therefore, I've created a workaround solution that will ensure the network adapters on Kali Linux receive an IPv4 address even after the upgrade. To resolve this issue, you can follow these steps:

1. Use the following command to download a custom script for ensuring IPv4 addresses are assigned to all network adapters on Kali Linux:

```
kali@kali:~$ wget https://raw.githubusercontent.com/PacktPublishing/The-Ultimate-Kali-Linux-Book-3E/main/Chapter%2002/ne
```

2. Next, use the following command to move the script to the `/etc/systemd/system/` services directory:

```
kali@kali:~$ sudo mv network-configuration.service /etc/systemd/system/
```

3. Next, reload `systemd` to load the new script as a service:

```
kali@kali:~$ sudo systemctl daemon-reload
```

4. Next, use the following command to enable the service to run at boot time:

```
kali@kali:~$ sudo systemctl enable network-configuration.service
```

Next, the PimpMyKali script from Dewalt enables us to both fix and install very useful utilities and tools that are commonly used by penetration testers and upgrade the existing software packages on our Kali Linux virtual machine. Keep in mind that you are running this script at your own risk on your Kali Linux machine. You can follow these instructions:

1. To run this script, open the Terminal and use the following commands by Dewalt:

```
kali@kali:~$ git clone https://github.com/Dewalt-arch/pimpmykali
kali@kali:~$ cd pimpmykali
kali@kali:~/pimpmykali$ sudo ./pimpmykali.sh
```

2. Next, the PimpMyKali command-line menu will appear with many options; enter `N` since we are running this script on a new virtual machine, as highlighted here:

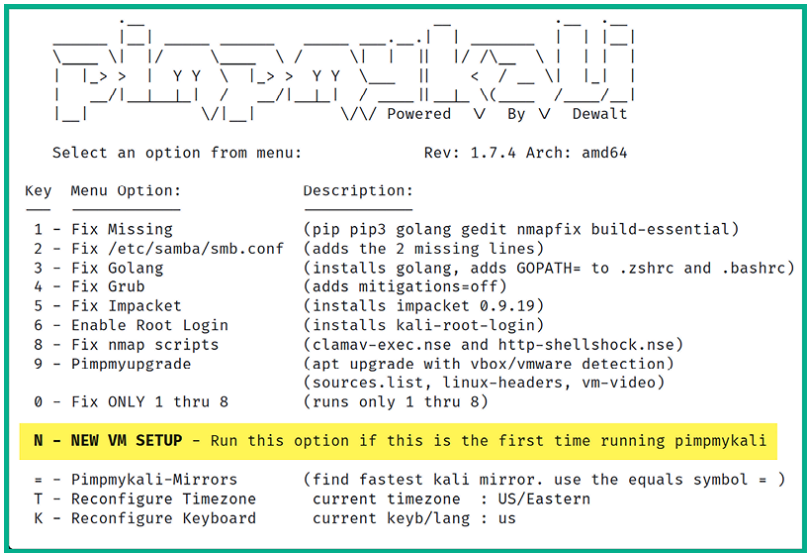


Figure 2.32: The PimpMyKali menu

During the setup process, the script will ask whether you want to re-enable the ability to log in with the `root` account on Kali Linux, but this is a personal preference. From a cybersecurity perspective, we should use an `administrator` or `root` account only when needed. I entered `N` (no) and hit *Enter* to continue the process. Keep in mind that this setup process takes a few minutes to complete.

3. Finally, after the setup process is completed, you will need to reboot the Kali Linux virtual machine to ensure all the configurations take effect. You will find the power options at the top-right corner of the Kali Linux desktop interface.

To learn more about Dewalt's PimpMyKali script, please see the official GitHub repository at **https://github.com/Dewalt-arch/pimpmykali**.

On March 29, 2024, it was reported that the `xz-utils` package, specifically version 5.6.0-0.2 on Linux-based systems, including Kali Linux, was vulnerable and contained a backdoor to potentially allow a threat actor to compromise the `sshd` authentication system, allowing a threat actor to gain unauthorized access to your Kali Linux machine. Use the following instructions to check the `xz-utils` package version:

1. The following command enables us to verify the installed version on Kali Linux:

```
kali@kali:~$ apt-cache policy liblzma5
```

This is shown in the following screenshot:

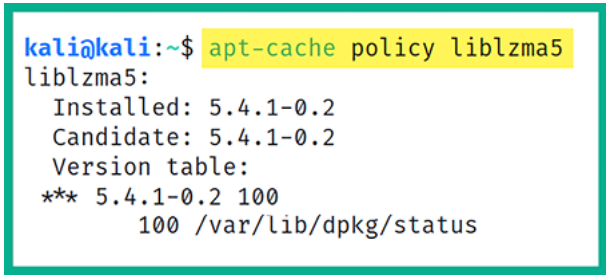


Figure 2.33: Checking for the vulnerable xz-utils package

2. To upgrade to the latest version, use the following command:

```
kali@kali:~$ sudo apt update && sudo apt install -y --only-upgrade liblzma5
```

After upgrading `liblzma5`, executing the `apt-cache policy liblzma5` command enables us to verify that Kali Linux is running the latest version at the time of writing:

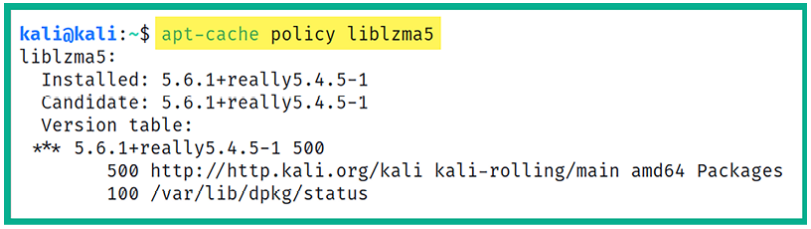


Figure 2.34: Verifying the upgrade of the xz-utils package

Note

To learn more about this security vulnerability, please see the official Kali Linux blog post at https://www.kali.org/blog/about-the-xz-backdoor/ and the National Vulnerability Database at https://nvd.nist.gov/vuln/detail/CVE-2024-3094.

Having completed this section, you have learned how to set up Kali Linux as a virtual machine, enable internet and other network connections for the virtual machine, and update the package repository source list. Next, you will learn how to set up a vulnerable web application to explore web application penetration testing in later sections of this book.

Setting up a vulnerable web application

Learning how to simulate real-world cyberattacks using Kali Linux would not be complete without understanding how to discover and exploit vulnerabilities within web applications. The **OWASP** is an organization that focuses on improving security through software, including web applications. The OWASP is known for its OWASP Top 10 list of most critical security risks within web applications. In *Chapters 16* and *17*, you will learn how to identify and exploit common vulnerabilities within web applications.

Note

At the time of writing this book, the latest version of the OWASP Top 10 was last updated in 2021. More information can be found at https://owasp.org/www-project-top-ten/. Further information on each of the Top 10 security risks is covered in *Chapters 16* and *17*.

As an aspiring ethical hacker and penetration tester, it's important to understand how to identify and perform security testing on each category within the OWASP Top 10 list. The OWASP created a few projects that allow learners to safely use their offensive security skills and techniques in a safe environment to discover web application vulnerabilities and exploit them. In this section, we'll be deploying the OWASP Juice Shop vulnerable web application on Kali Linux.

To get started with setting up the OWASP Juice Shop web application, please use the following instructions:

1. Firstly, power on your **Kali Linux virtual machine** and log in.

2. Next, open the **Terminal** and use the following commands to update the package repository list and install Docker:

```
kali@kali:~$ sudo apt update
kali@kali:~$ sudo apt install -y docker.io
kali@kali:~$ sudo systemctl start docker
kali@kali:~$ sudo systemctl enable docker
```

Use the `docker --version` command to test whether Docker is installed correctly on Kali Linux.

3. Next, use the installed Docker application to pull the **OWASP Juice Shop** container from the online Docker Hub repository:

```
kali@kali:~$ sudo docker pull bkimminich/juice-shop
```

The following screenshot shows the download and setup process of the OWASP Juice Shop Docker container:

```
kali@kali:~$ sudo docker pull bkimminich/juice-shop
Using default tag: latest
latest: Pulling from bkimminich/juice-shop
383e1c5dd0c1: Pull complete
c59673e9fae3: Pull complete
7dcffaf98769: Pull complete
110615d32fe3: Pull complete
aa52b96be1e2: Pull complete
15e0f40066fa: Pull complete
Digest: sha256:073163e118541daec3a26321d6fb70e7454ab369de5f296c131f5ff99fc8c91c
Status: Downloaded newer image for bkimminich/juice-shop:latest
docker.io/bkimminich/juice-shop:latest
```

Figure 2.35: Downloading the Juice Shop Docker container

4. Next, use the following command to run the OWASP Juice Shop Docker container on port `3000`:

```
kali@kali:~$ sudo docker run --rm -p 3000:3000 bkimminich/juice-shop
```

The following snippet shows the execution of the preceding command:

```
kali@kali:~$ sudo docker run --rm -p 3000:3000 bkimminich/juice-shop
info: All dependencies in ./package.json are satisfied (OK)
info: Detected Node.js version v18.15.0 (OK)
info: Detected OS linux (OK)
info: Detected CPU x64 (OK)
info: Configuration default validated (OK)
info: Entity models 19 of 19 are initialized (OK)
info: Required file server.js is present (OK)
info: Required file index.html is present (OK)
info: Required file main.js is present (OK)
info: Required file tutorial.js is present (OK)
info: Required file polyfills.js is present (OK)
info: Required file styles.css is present (OK)
info: Required file runtime.js is present (OK)
info: Required file vendor.js is present (OK)
info: Port 3000 is available (OK)
info: Chatbot training data botDefaultTrainingData.json validated (OK)
info: Server listening on port 3000
```

Figure 2.36: Running the Juice Shop Docker container

Note

To stop the Docker container from running, use the *CTRL + C* key combination.

5. Next, open the Firefox web browser within Kali Linux and go to `http://127.0.0.1:3000` to access and interact with the OWASP Juice Shop web application, as shown in the following screenshot:

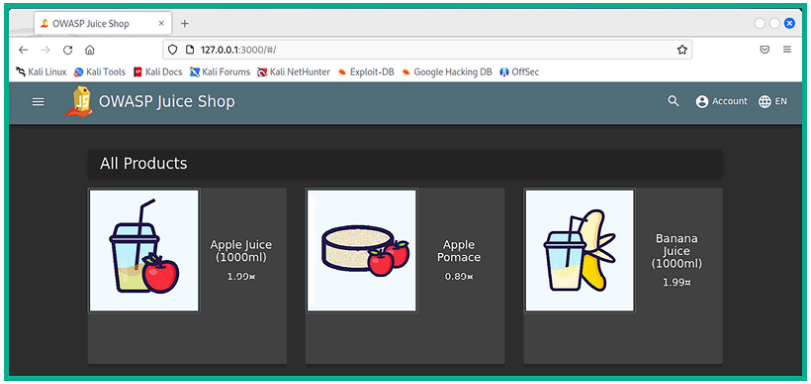


Figure 2.37: The Juice Shop homepage

To learn more about the OWASP Juice Shop vulnerable web application, please see the official documentation at https://owasp.org/www-project-juice-shop/.

6. Lastly, use the following commands to stop the Docker service:

```
kali@kali:~$ sudo systemctl stop docker
kali@kali:~$ sudo systemctl disable docker
```

To verify the status of the Docker service, use the `sudo systemctl status docker` command. In addition, if you having network connectivity issues to the internet from Kali Linux, simply reboot the virtual machine.

Having completed this exercise, you have learned how to set up Docker and the OWASP Juice Shop on Kali Linux. Next, you will learn how to set up Metasploitable 2, a vulnerable Linux-based system in our lab environment.

Deploying Metasploitable 2 as a vulnerable machine

When building a penetration testing lab, it's important to include vulnerable systems that will act as our targets. These systems contain intentionally vulnerable services and applications, enabling us to practice and build our skills to better understand how to discover and exploit vulnerabilities. A very popular vulnerable machine is known as Metasploitable 2. This vulnerable machine contains a lot of security vulnerabilities that can be exploited and is good for learning about ethical hacking and penetration testing.

To get started setting up Metasploitable 2 within our lab environment, please use the following instructions:

Part 1 – deploying Metasploitable 2

The following steps will guide you to acquiring the Metasploitable 2 virtual machine and deploying it within Oracle VM VirtualBox Manager:

1. Firstly, on your host computer, go to https://sourceforge.net/projects/metasploitable/files/Metasploitable

2/ to download the `metasploitable-linux-2.0.0.zip` file onto your device. Once the ZIP file has been downloaded, extract (unzip) its contents. The extracted files are the virtual hard disk and settings configuration files for the Metasploitable 2 virtual machine.

2. Next, let's create a virtual machine for Metasploitable 2, open **Oracle VM VirtualBox Manager**, and click on **New**.
3. When the **Create Virtual Machine** window appears, click on **Expert Mode** to change the configuration view.
4. Next, within the **Name and Operating System** section, use the following configurations for the virtual machine:
   1. **Name**: `Metasploitable 2`
   2. **Type**: **Linux**
   3. **Version**: **Other Linux (64-bit)**

   The following screenshot shows the preceding settings on the **Create Virtual Machine** window:

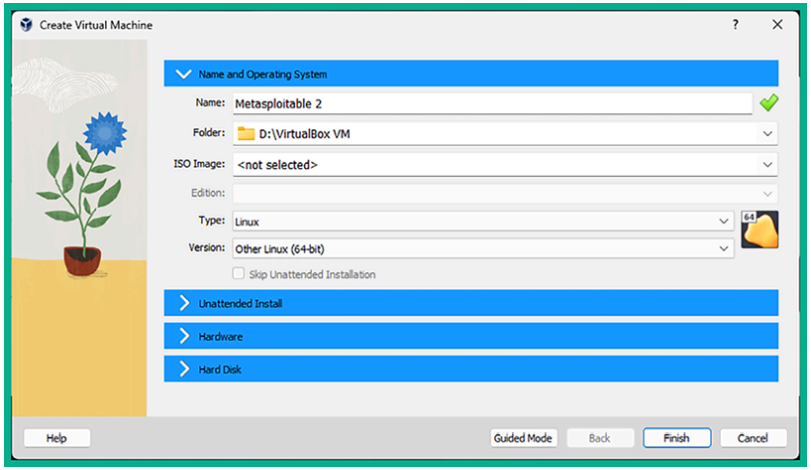


Figure 2.38: The Create Virtual Machine window

5. Next, expand the **Hard Disk** category on the **Create Virtual Machine** window, select the **Use an Existing Virtual Hard Disk File** option, and then click on the folder icon on the right side:

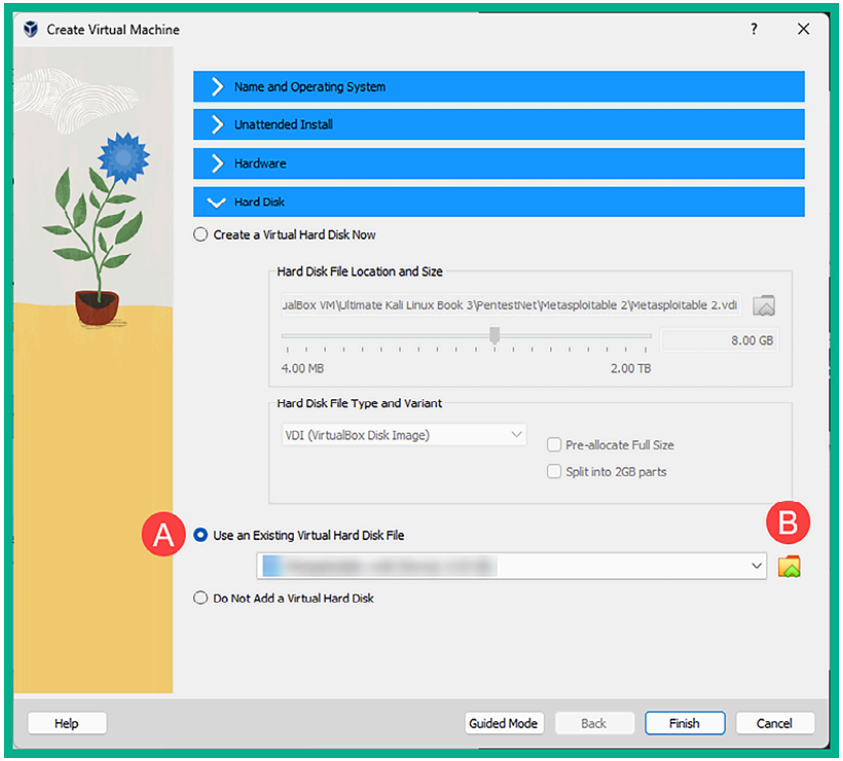


Figure 2.39: Virtual hard disk settings

6. Next, in the **Hard Disk Selector** window, click on **Add**:

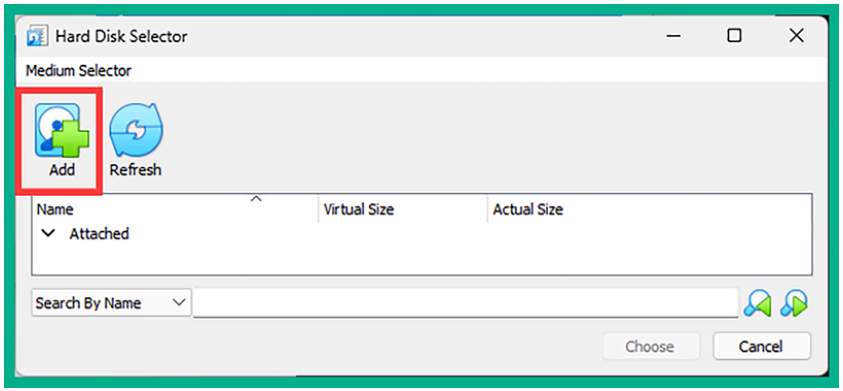


Figure 2.40: Adding an existing virtual hard disk

7. Next, a pop-up window will appear; use it to navigate to the **Metasploitable 2** extracted folder and its contents, select the **Metasploitable** VMDK file and click on **Open**, as shown here:

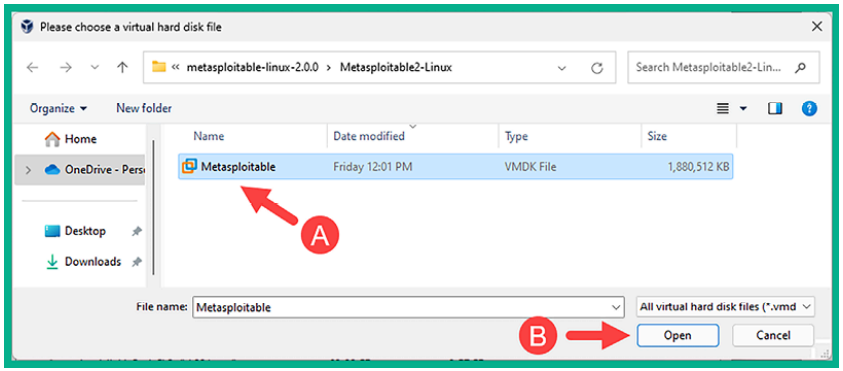


Figure 2.41: Selecting the virtual hard disk

8. Next, you will automatically return to the **Hard Disk Selector** window where the **Metasploitable** disk file will be available; select it and click on **Choose**:

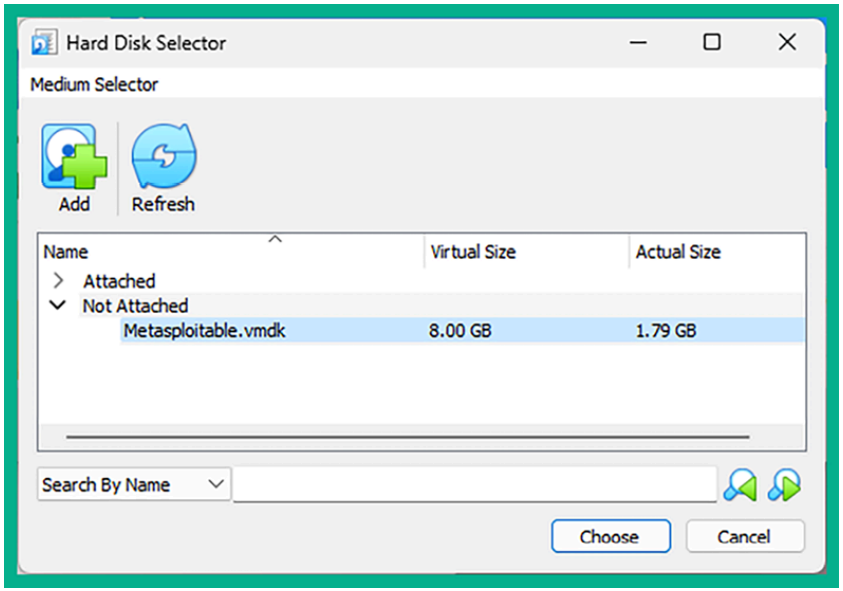


Figure 2.42: Choosing the imported virtual hard disk

9. Next, you'll automatically return to the **Create Virtual Machine** window where you'll see the `Metasploitable.vmdk` file is loaded as the existing virtual disk file. Here, click on **Finish**:

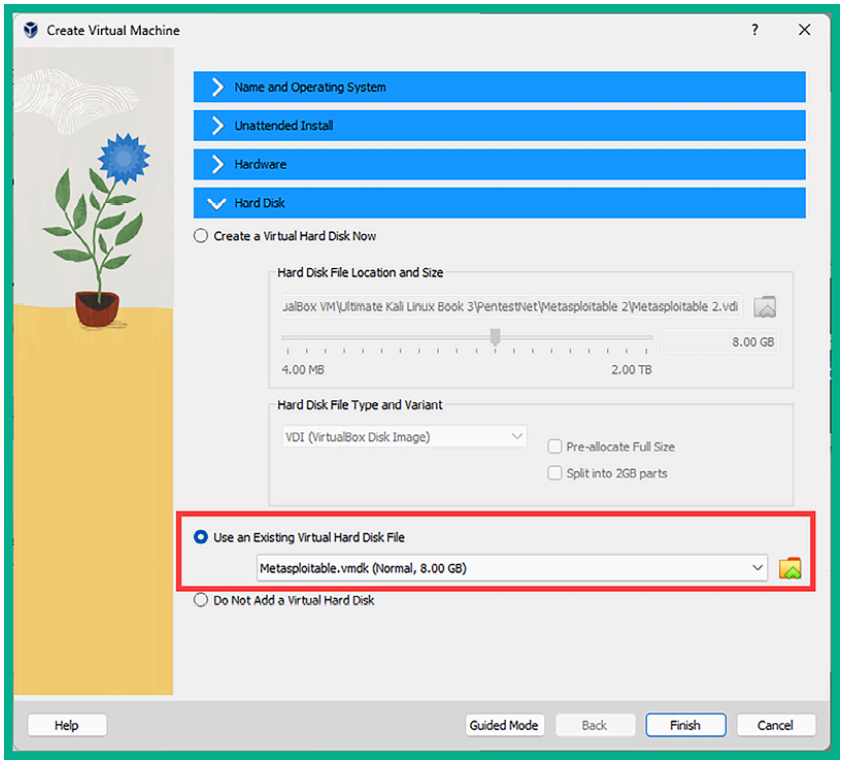


Figure 2.43: Using the newly imported virtual hard disk

At this point, the Metasploitable 2 virtual machine is created and loaded within the Oracle VM VirtualBox Manager. Next, we will connect the Metasploitable 2 virtual machine to the *PentestNet* virtual network.

Part 2 – configuring network settings

Since our penetration testing lab topology contains more than one virtual network, the following steps will help ensure Kali Linux has end-to-end network connectivity with the Metasploitable 2 virtual machine:

1. To configure the networking settings, select the newly created **Metasploitable 2** virtual machine within **Oracle VM VirtualBox Manager** and click on

Settings.

2. Next, go to the **Network** | **Adapter 1** and use the following configurations:
    1. Enable the network adapter
    2. **Attached to**: **Internal Network**
    3. **Name**: `PentestNet` (manually type it in the field)
    4. **Promiscuous Mode**: **Allow All**

    The following screenshot shows the preceding configurations on **Adapter 1**. Click **OK** to save:

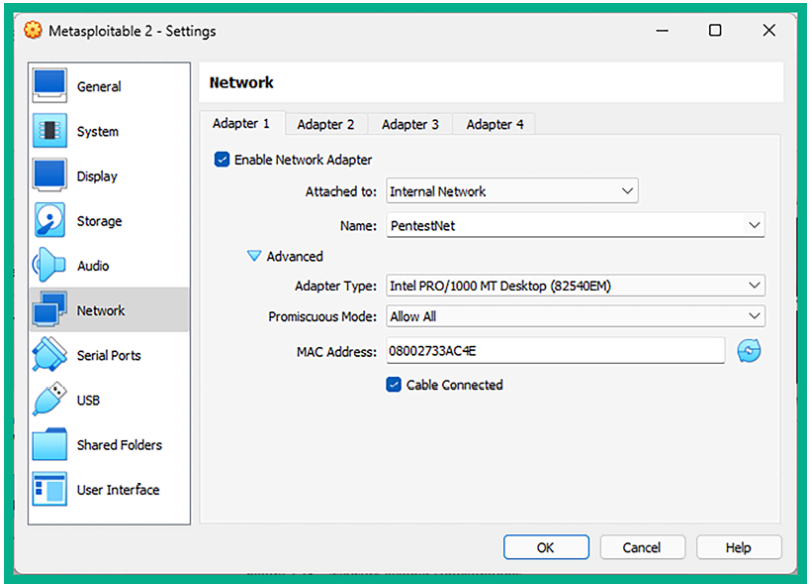


Figure 2.44: Network Adapter 1 settings

3. Next, power on the **Metasploitable 2** virtual machine and log in using the username: `msfadmin` and password: `msfadmin`. Then, use the `ip address` command to verify the virtual machine is receiving an IP address on the `172.30.1.0/24` network:

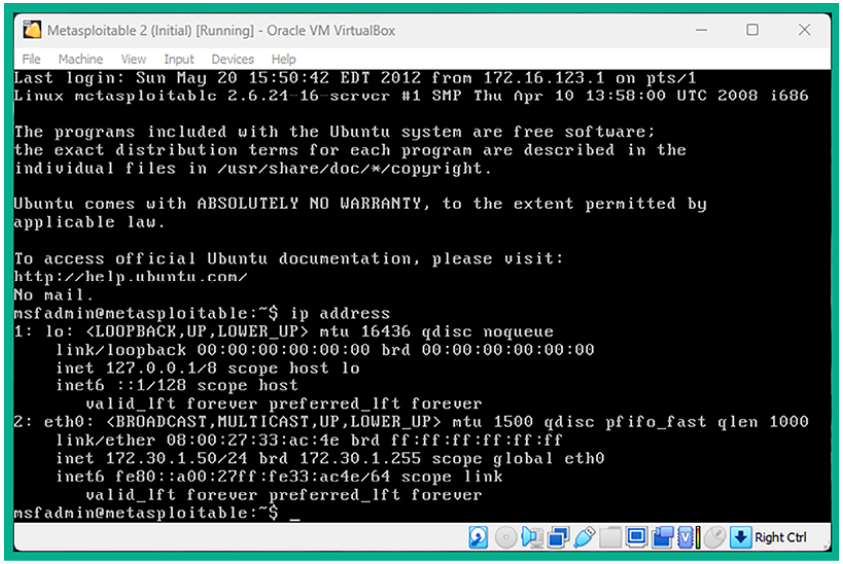


Figure 2.45: Verifying IP assignment

If your mouse cursor is stuck within a virtual machine, press the right *Ctrl* key to detach the cursor.

4. Lastly, use the `sudo halt` command to power off the Metasploitable 2 virtual machine.

Having completed this section, you have learned how to set up Metasploitable 2 as a vulnerable machine within our penetration testing lab. Next, you will learn how to build and deploy Metasploitable 3 using Vagrant.

Building and deploying Metasploitable 3

In this section, you will learn how to build and deploy Metasploitable 3, both the Windows server and Linux server versions. The Windows server version will be using a dual-homed network connection to both the *PentestNet* network (`172.30.1.0/24`) and *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 2.46: Low-level lab diagram

As shown in the preceding diagram, this topology goes more in 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 building and deploying Metasploitable 3 – Windows version, please follow these instructions:

1. Firstly, you will need to download and install **Vagrant** on your host computer. Vagrant enables users to both build and maintain virtual machines and applications. On your host computer, go to `https://www.vagrantup.com/` and click on the **Download** button on the web page.
2. Next, select and download **Vagrant AMD64 version 2.3.7** as shown in the following screenshot:

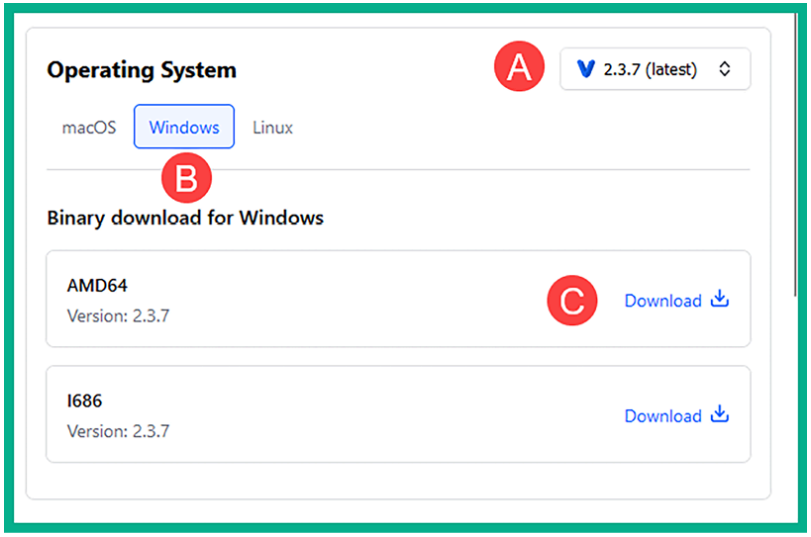


Figure 2.47: The Vagrant download page

After downloading the Vagrant software package, double-click on the installer package to start the installation process. After the installation is completed, you'll be prompted to reboot your host computer to ensure the changes are effective.

3. After your host computer reboots, open the Windows **Command Prompt** and use the following commands to reload and install additional plugins for Vagrant:

```
C:\Users\Glen> vagrant plugin install vagrant-reload
C:\Users\Glen> vagrant plugin install vagrant-vbguest
```

The following screenshot shows the execution of the preceding commands:

Figure 2.48: Vagrant commands

4. Next, use the following commands to load the Metasploitable 3 – Windows server version to your system using Vagrant:

```
C:\Users\Glen> vagrant box add rapid7/metasploitable3-win2k8
```

5. Next, select option `1` to use **VirtualBox** as the preferred hypervisor:

Figure 2.49: Selecting the Metasploitable 3 Vagrant image

6. Vagrant will begin to download the virtual machine files for the Metasploitable – Windows version as shown here:

Figure 2.50: Downloading the Metasploitable 3 image

7. Next, change the current working directory to `.vagrant.d\boxes`, rename the `rapid7-VAGRANTSLASH-metasploitable3-win2k8` folder, and initialize the

build configurations for the Metasploitable 3 – Windows virtual machine using the following commands:

```
C:\Users\Glen> cd .vagrant.d\boxes
C:\Users\Glen\.vagrant.d\boxes> REN "rapid7-VAGRANTSLASH-metasploitable3-win2k8" "metasploitable3-win2k8"
C:\Users\Glen\.vagrant.d\boxes> vagrant init metasploitable3-win2k8
```

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

Figure 2.51: Initializing the Metasploitable 3 image

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

```
C:\Users\Glen\.vagrant.d\boxes> vagrant up
```

The following screenshot shows the execution of the preceding commands:

Figure 2.52: Building and setting up the Metasploitable 3 image

This process usually takes a few minutes to complete.

If the `vagrant up` command gives an error, execute it again.

9. After the process is completed, open the **Oracle VM VirtualBox Manager**. Here, you will find a newly created virtual machine named `boxes_default_*` is running. This is the Metasploitable 3 – Windows virtual machine. Select it and click on **Show**:

Figure 2.53: VirtualBox with Metasploitable 3

10. 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 2.54: Input menu on VirtualBox

11. Select the **Administrator** account and use the default password: `vagrant` to log in, as shown here:

Figure 2.55: Metasploitable 3 login screen

Once you're logged in, simply close all the windows that appear and do not activate the operating system.

12. Click on the **Start** icon in the bottom-left corner and select the **Shutdown** button to shut down / turn off the operating system.
13. Next, on the **Oracle VM VirtualBox Manager**, select the **Metasploitable 3 – Windows** virtual machine and click on **Settings**.
14. Then, select the **Network** category and use the following configurations for **Adapter 1**:
    1. Enable the network adapter
    2. **Attached to**: **Internal Network**
    3. **Name**: `PentestNet` (manually type it in the field; it is case sensitive)
    4. **Promiscuous Mode**: **Allow All**

    The following screenshot shows the preceding configurations on **Adapter 1**:

Figure 2.56: Network | Adapter 1

15. Next, select **Adapter 2** and use the following configurations:
    1. Enable the network adapter
    2. **Attached to**: **Internal Network**
    3. **Name**: `HiddenNet` (manually type it in the field; it is case sensitive)
    4. **Promiscuous Mode**: **Allow All**

    The following screenshot shows the preceding configurations on **Adapter 2**:

Figure 2.57: Network | Adapter 2

To rename your virtual machine, navigate to **Settings** | **General** | **Basic** | **Name**.

16. Lastly, ensure Kali Linux has end-to-end connectivity with the Metasploitable 3 – Windows virtual machine on the network.

Next, you will deploy Metasploitable 3 – Linux virtual machine within the HiddenNet network.

Part 2 – building the Linux server version

To start setting up the Linux version of Metasploitable 3 within our lab environment, please follow these instructions:

1. On the Windows **Command Prompt**, use the following commands to load the Linux version of Metasploitable 3 on your host device using Vagrant:

```
C:\Users\Glen\.vagrant.d\boxes> vagrant box add rapid7/metasploitable3-ub1404
```

2. Next, choose option **1** and hit *Enter* to download the virtual machine files for Metasploitable 3 – Linux version, as shown here:

Figure 2.58: Adding the Linux version of Metasploitable 3

3. Next, delete the `Vagrantfile` file, rename the `rapid7-VAGRANTSLASH-metasploitable3-ub1404` folder, and initialize the build configurations for the Metasploitable 3 – Linux virtual machine using the following commands:

```
C:\Users\Glen\.vagrant.d\boxes> del Vagrantfile
C:\Users\Glen\.vagrant.d\boxes> REN "rapid7-VAGRANTSLASH-metasploitable3-ub1404" "metasploitable3-ub1404"
C:\Users\Glen\.vagrant.d\boxes> vagrant init metasploitable3-ub1404
```

The following screenshot shows the execution of the preceding commands:

Figure 2.59: Initializing the Linux version of Metasploitable 3

You may need to open **Oracle VM Virtual Manager** before proceeding to the next step. If you do not, the next step may not work correctly.

4. Next, open **Windows Explorer** and go to `C:\Users\<userrname>\.vagrant.d\boxes\metasploitable3-ub1404\0.1.12-weekly\virtualbox`, where you will find the compiled virtual machine files. Right-click on the **box** file and click **Open with** | **VirtualBox Manager**:

Figure 2.60: Importing Metasploitable 3 (Linux) into VirtualBox

5. Next, the **Import Virtual Appliance** window will appear. Click on the **Finish** button shown in the following screenshot:

Figure 2.61: Metasploitable 3 (Linux) appliance settings

6. Next, the **metasploitable3-ub1404** virtual machine will be imported on **Oracle VM VirtualBox Manager**. Select it and click on **Settings**:

Figure 2.62: VirtualBox with Metasploitable 3 (Linux version)

7. Next, select **Adapter 1** and use the following configurations:
    1. Enable the network adapter
    2. **Attached to**: **Internal Network**
    3. **Name**: `HiddenNet` (manually type it in the field)
    4. **Promiscuous Mode**: **Allow All**

    The following screenshot shows the preceding configurations on **Adapter 1**:

Figure 2.63: Network | Adapter 1

8. Finally, power on the **metasplotable3-ub1404** virtual machine and log in using the username: `vagrant` and password: `vagrant`. Once you're logged in, use the `ip address` command to verify the virtual machine is receiving an IP address on the `10.11.12.0/24` network:

Figure 2.64: Verifying IP assignment

Use the `sudo halt` command to power off this virtual machine.

Having completed this section, you have learned how to set up both versions of Metasploitable 3 within your lab environment. Metasploitable 3 contains newer vulnerabilities than its predecessor and will be fun to exploit in later chapters of this book.

Summary

Having completed this chapter, you learned about the importance of building your very own penetration testing lab on your computer. You learned how to use hypervisors to virtualize the hardware resources on a system, which can then be shared with multiple operating systems that are running at the same time on the same system. In addition, you have gained the skills of setting up and deploying Kali Linux, multiple vulnerable systems, and web applications within a virtualized environment.

You established a foundational understanding of virtualization technology, gained practical experience in configuring a secure, isolated lab environment, and practiced hands-on skills in utilizing penetration testing tools within that environment.

I trust that the knowledge presented in this chapter has provided you with valuable insights, supporting your path toward becoming an ethical hacker and penetration tester in the dynamic field of cybersecurity. May this newfound understanding empower you in your journey, allowing you to navigate the industry with confidence and make a significant impact. In the next chapter, you will learn how to set up an AD lab environment for performing red teaming techniques used in later chapters.

Further reading

- OWASP Top 10 – https://owasp.org/www-project-top-ten/
- Kali Linux Blog – https://www.kali.org/blog/

Join our community on Discord

Join our community's Discord space for discussions with the author and other readers:

https://packt.link/SecNet