6
Active Reconnaissance

The more information is collected about a target, the more it helps ethical hackers and penetration testers improve exploit development during the weaponization phase of the Cyber Kill Chain and identify the best method to deliver the malicious payload to the target. Active reconnaissance helps you collect information that's not publicly available, such as which services are running and how many ports exist on a targeted system. For instance, if you're targeting a web server, it's important to identify the web application and its version. In addition, it would be useful to also identify the operating system that's hosting the web application.

During this chapter, you will understand the need for **active reconnaissance** techniques during ethical hacking and penetration testing assessments on a target system, network, and organization. You will explore **active scanning** techniques, which are commonly used to identify live systems, their open port, and running services. Using **fingerprinting** techniques, penetration testers can identify the operating systems, service versions of running services, and configurations on a targeted system, which helps you profile the target and identify their attack surface, which can help improve their plan of attack. Lastly, you will learn how to perform enumeration on common network services and identify whether an organization is leaking data on its cloud platform.

In this chapter, we will cover the following topics:

- Understanding active information
- Profiling websites using EyeWitness
- Exploring active scanning techniques
- Using scanning evasion techniques
- Enumerating common network services
- Discovering data leaks in the cloud

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/
- EyeWitness – https://github.com/RedSiege/EyeWitness
- S3Scanner – https://github.com/sa7mon/S3Scanner

If you're unable to connect to the internet from Kali Linux, use the `cat /etc/resolv.conf` command to determine whether your DNS servers are set correctly on Kali Linux, then use the `sudo systemctl restart NetworkManager` command to restart the Network Manager stack. As a last resort, you can restart the Kali Linux operating system.

Understanding active information

Using active reconnaissance techniques enables ethical hackers and penetration testers to use a more direct approach when engaging the target. For instance, many active reconnaissance techniques involve establishing a logical network connection between your attacker machines, such as Kali Linux, and the targeted systems over the network. With active reconnaissance, you can send specially crafted probes to collect specific details, for example, by doing the following:

- Determining how many live hosts are on a network
- Determining whether the targeted system is online
- Identifying open port numbers and running services
- Profiling the operating system on the targeted machine
- Identifying whether the targeted system has any network shares

Therefore, before launching any type of network-based attack, it's important to determine whether there are live systems on the network and whether the target is online. Imagine launching an attack toward a specific system, only to realize the target is offline and the attack has failed. Hence, it doesn't make sense to target an offline device as it would be unresponsive and increase the risk of detection by the organization's security team.

Unlike passive reconnaissance, which leverages **open-source intelligence (OSINT)** from public data sources, using active reconnaissance techniques does increase the risk of being detected by the target's se-

curity systems and triggering alerts. Therefore, it's important to consider the threat level for each type of attack during your planning phase.

Compared to adversaries, ethical hackers and penetration testers use similar techniques to simulate a real-world cyberattack to identify how a real attacker would collect and leverage information to identify security vulnerabilities and compromise their targets.

In the next section, you will learn how to automate the process of taking screenshots of targeted domains and systems on a network.

Profiling websites using EyeWitness

What do you do after discovering additional sub-domains of a targeted organization on the internet? A common and obvious practice would be to visit each sub-domain to determine whether it leads to a vulnerable web application or system that can be exploited to gain a foothold in the targeted organization's network.

However, manually visiting each sub-domain can be quite time-consuming if you need to visit 100+ sub-domains for a targeted organization. As an aspiring ethical hacker and penetration tester, using a tool such as **EyeWitness** enables you to automate the process of checking and capturing a screenshot of each sub-domain. EyeWitness also has the capability of analyzing the response headers from HTTP messages and identifying default credentials in known login pages on a web application.

To get started using EyeWitness, please use the following instructions:

1. Power on the **Kali Linux** virtual machine, open the Terminal, and use the following command to clone the EyeWitness repository:

```
kali@kali:~$ git clone https://github.com/RedSiege/EyeWitness
```

2. Next, execute the `setup.py` script to install EyeWitness by using the following commands:

```
kali@kali:~$ cd EyeWitness/Python/setup
```

```
kali@kali:~/EyeWitness/Python/setup$ sudo ./setup.sh
```

Next, use the `cd ..` command to move up one directory, as shown here:

```
kali@kali:~/EyeWitness/Python/setup$ cd ..
```

3. Next, use the following commands to first create a new text file within the `/home/kali/` directory, then write a targeted sub-domain into it:

```
kali@kali:~/EyeWitness/Python$ touch /home/kali/eyewitness_targets.txt
kali@kali:~/EyeWitness/Python$ echo https://example.com/ > /home/kali/eyewitness_targets.txt
```

The `touch <filename>` command enables you to create a new file within Linux. The `echo` command allows you to write contents within a file.

4. Next, use the following command to enable EyeWitness to capture screenshots of each sub-domain found within the `eyewitness_targets.txt` file:

```
kali@kali:~/EyeWitness/Python$ ./EyeWitness.py --web -f /home/kali/eyewitness_targets.txt -d /home/kali/EyeWitness_Scr
```

The following is a breakdown of each syntax used in the preceding command:

1. `--web` : This specifies to take HTTP screenshots.
2. `-f` : This specifies the source file with the list of targeted domains and sub-domains.
3. `-d` : This specifies the output directory to save the results and report.
4. `--prepend-https` : This specifies to prepend `http://` and `https://` to the list of domains and sub-domains.

The following screenshot shows the process of capturing the screenshots:

```
################################################################################
#                                 EyeWitness                                   #
################################################################################
#           Red Siege Information Security - https://www.redsiege.com          #
################################################################################

Starting Web Requests (1 Hosts)
Attempting to screenshot https://example.com/
Finished in 4.101680278778076 seconds

[*] Done! Report written in the /home/kali/EyeWitness_Screenshots folder!
Would you like to open the report now? [Y/n]
```

Figure 6.1: Capturing screenshot

5. If you type `Y` and hit *Enter*, the EyeWitness report will automatically load and open within the web browser, as shown here:

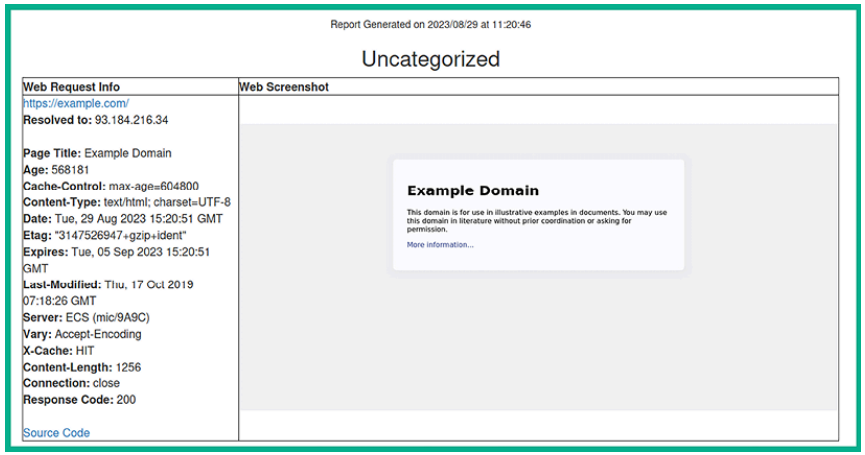


Figure 6.2: EyeWitness report

As you have seen, using a tool such as EyeWitness can save you a lot of time as compared to checking each sub-domain manually. You can quickly browse each image within the generated report to identify any login portals and sensitive directories on a targeted domain.

To learn more about EyeWitness, please see **https://github.com/RedSiege/EyeWitness** and use the `./EyeWitness.py -h` command to view the help menu.

Having completed this section, you have learned how to automate the process of capturing screenshots of many websites using EyeWitness. In the next section, you will explore various scanning and fingerprinting techniques.

Exploring active scanning techniques

As an aspiring ethical hacker and penetration tester, it's essential to develop a solid foundation on understanding how to leverage active scanning techniques to efficiently discover and profile targeted systems on an organization's network. Unlike passive reconnaissance, active reconnaissance focuses on sending special probes directly to a targeted system to retrieve specific information, which isn't available from OSINT data source. In addition, active scanning helps us identify accurate information about the target, while some OSINT data sources may not have the latest version of the information.

Many organizations focus on securing their perimeter network and sometimes do not apply equal focus on securing their internal network (of the cyberattacks I've encountered in my career, 90% usually originate from inside the network). Due to this, many organizations think the attacker will launch their attack from the internet, which will then be blocked by their network-based firewall.

The following diagram shows a simplified overview of a typical deployment of a firewall:

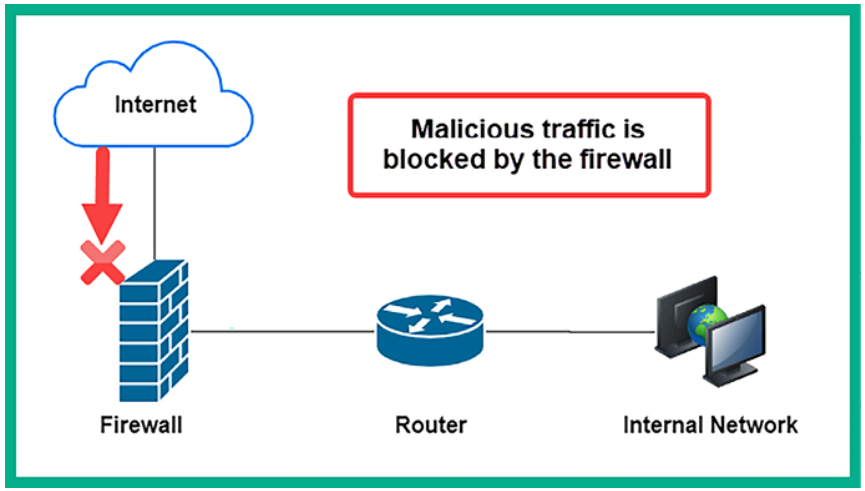


Figure 6.3: Network-based firewall

As shown in the preceding diagram, the network-based firewall is implemented as the edge device between the organization's internal network and the internet. One of its roles is to filter traffic between different networks and prevent malicious traffic from passing through. This includes blocking malicious traffic from the internet intended for internal systems on the organization's network and vice versa.

However, threat actors are continuously learning how organizations implement their infrastructure and security solutions, as well as the decisions that both the leadership team and IT professionals make when securing their assets.

While many organizations are investing in their cyber defenses to ensure their assets and people are protected from adversaries and cyberattacks, there are still so many organizations around the world without firewalls, misconfigured network devices and security appliances, and unpatched operating systems. Metaphorically speaking, it's only a matter of time before an adversary discovers this gold mine and starts *living off the land*. During the reconnaissance phase of the Cyber Kill Chain and common penetration testing methodology, ethical hackers and penetration testers will eventually need to directly engage with the target to collect information that is not available from OSINT and use active reconnaissance techniques such as scanning and enumeration.

Scanning is a technique that's used by threat actors to discover live systems on a network, identify the open service ports on a system, and discover vulnerabilities on host machines and even their operating system architecture. The information that's gathered from scanning helps the penetration tester gain a clearer view of their targets compared to passive information gathering.

Do not perform any type of scanning on systems and networks that you do not own or have legal permission to do so. Scanning is considered illegal in many countries.

Penetration testers always need to improve their critical thinking mindset to think like a real threat actor, especially if they want to perform a successful penetration test on a targeted organization. In addition, they should develop problem-solving skills, analytical thinking, creativity in bypassing security measures, and adaptability to the evolving security landscape. In the following subsections, you

will learn about various techniques and methodologies for performing scanning on a targeted network and how to profile systems.

Changing your MAC address

The **Network Interface Card** (**NIC**) is a network adapter that enables a system to communicate over a wired or wireless network. For instance, before your devices send data on a network, the NIC converts the message into a signal that's supported over the media for transmission, such as electrical signals for copper cables, light signals for fiber optics, and radio frequency for wireless communication. In addition, the NIC on each device contains a globally unique **Media Access Control** (**MAC**) address, sometimes referred to as a *burned-in address*, that's theoretically not changeable.

Before a device transmits data over a network, the sender device automatically inserts the source and destination MAC address onto the frame header of the message. The source MAC address helps the recipient identify the sender of the message, and the destination MAC address helps the network switch forward the message to the intended destination. However, if the destination MAC address is unknown by the sender device, the sender device will broadcast an **Address Resolution Protocol** (**ARP**) request message to the network. Only the device with the targeted MAC address will respond, providing its MAC address.

The **Neighbor Discovery Protocol** (**NDP**) is used on IPv6 networks for address resolution.

The MAC address is a 48-bit address written in hexadecimal. The first 24 bits of the address are known as the **Organizationally Unique Identifier** (**OUI**), which helps IT professionals determine the vendor of a device, while the last 24 bits are uniquely assigned by the vendor. Therefore, when your NIC sends traffic out on a network, your real MAC address is also inserted within the frame header, and this information can be used to identify your machine on a network.

As an aspiring penetration tester, you can change the MAC address on both your Ethernet and wireless network adapters by using a pre-installed tool known as **MAC Changer**. Changing your MAC address allows you to trick other devices on the network into thinking your system is a common device that belongs within the organization's network infrastructure, such as a network device, a printer, or

a vendor-specific device. This technique is commonly used to protect the identity of your attacker machine, bypass MAC filtering rules on network devices, and evade network restrictions while on your target's network.

While changing a MAC address can evade some network restrictions, it is not a foolproof method for anonymity or bypassing security measures. Network monitoring tools can detect anomalies in traffic patterns.

To learn how to change your MAC address using **MAC Changer**, please use the following instructions:

1. Power on the **Kali Linux** virtual machine and use the `ifconfig` command to determine the original MAC address on your network adapters, as shown here:

```
kali@kali:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:d1:[redacted]  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.17.15  netmask 255.255.255.0  broadcast 172.16.17.255
        ether 08:00:27:[redacted]  txqueuelen 1000  (Ethernet)
        RX packets 9  bytes 7020 (6.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 15  bytes 4409 (4.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 6.4: Checking network interfaces

As shown in the preceding screenshot, the `ifconfig` command was used to display all of the connected network adapters on the Kali Linux virtual machine. In addition, this command enables us to view the original MAC address on each network adapter, within the `ether` field.

2. Next, logically turn down the `eth0` interface with the following commands:

```
kali@kali:~$ sudo ifconfig eth0 down
```

3. Next, use the `macchanger --help` command to view a list of available options, as shown here:

```
kali@kali:~$ macchanger --help
GNU MAC Changer
Usage: macchanger [options] device

  -h,  --help                  Print this help
  -V,  --version               Print version and exit
  -s,  --show                  Print the MAC address and exit
  -e,  --ending                Don't change the vendor bytes
  -a,  --another               Set random vendor MAC of the same kind
  -A                           Set random vendor MAC of any kind
  -p,  --permanent             Reset to original, permanent hardware MAC
  -r,  --random                Set fully random MAC
  -l,  --list[=keyword]        Print known vendors
  -b,  --bia                   Pretend to be a burned-in-address
  -m,  --mac=XX:XX:XX:XX:XX:XX
       --mac XX:XX:XX:XX:XX:XX  Set the MAC XX:XX:XX:XX:XX:XX
```

Figure 6.5: MAC Changer options

4. Next, set a randomized MAC address on the `eth0` network adapter by using the following command:

```
kali@kali:~$ sudo macchanger -A eth0
```

The following screenshot shows the current, permanent, and the newly generated MAC addresses for the `eth0` network adapter:

```
kali@kali:~$ sudo macchanger -A eth0
Current MAC:   08:00:27:53:0c:ba (CADMUS COMPUTER SYSTEMS)
Permanent MAC: 08:00:27:53:0c:ba (CADMUS COMPUTER SYSTEMS)
New MAC:       00:18:f2:28:80:71 (Beijing Tianyu Communication Equipment Co., Ltd)
```

Figure 6.6: Changing MAC address

5. Next, re-enable the `eth0` interface by using the following command:

```
kali@kali:~$ sudo ifconfig eth0 up
```

6. Next, use the `ifconfig` command once more to verify that `eth0` has a spoofed MAC address, as shown here:

```
kali@kali:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.17.59  netmask 255.255.255.0  broadcast 172.16.17.255
        ether 00:18:f2:28:80:71  txqueuelen 1000  (Ethernet)
        RX packets 41  bytes 10264 (10.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 21  bytes 5585 (5.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 6.7: Using the ifconfig eth0 command for verification

7. Lastly, to further verify the vendor of the spoofed MAC address, go to https://macvendors.com/ and enter the MAC address, as shown here:

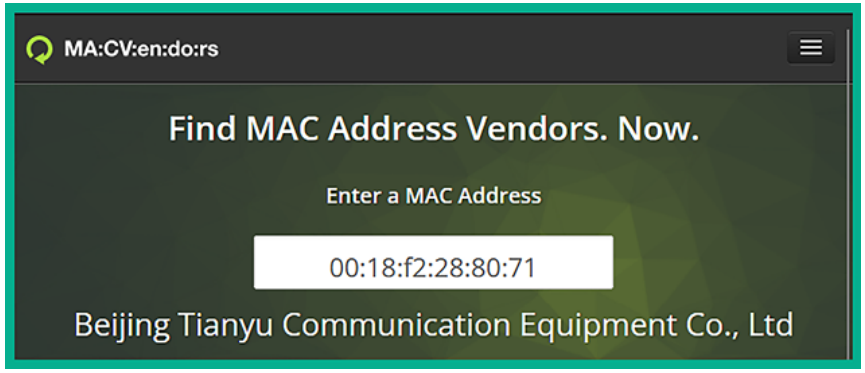


Figure 6.8: Verifying the vendor

Having completed this exercise, you have learned how to spoof your MAC address on Kali Linux. However, it's important to consider using a MAC address that's associated with a common vendor of networking devices or systems to reduce the risk of detection by the organization's security team. Next, you will learn how to perform host discovery to identify live systems on an internal network.

Performing live host discovery

Discovering live hosts on a targeted network is an essential stage when performing a penetration test. Let's imagine you're an ethical hacker or a penetration

tester; your targeted organization permits you to directly connect your attacker's machine with Kali Linux on their network to perform security testing on their internal network. You're eager to start discovering security vulnerabilities and hacking systems, but you're not sure whether the targeted hosts are online.

In this section, you will learn about the skills you will need to perform various types of active reconnaissance on an organization's networks using various tools and techniques. However, to ensure you can perform these exercises in a safe space, please use the following guidelines:

- Ensure you do not scan systems that you do not own or have been granted legal permission.
- Ensure the network adapter of Kali Linux is assigned to the **PentestNet** network within Oracle VM VirtualBox Manager.
- The PentestNet network will be our simulated organization network.

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

1. Power on the **Kali Linux**, **Metasploitable 2**, and **Metasploitable 3** (**Windows version**) virtual machines.
2. On **Kali Linux**, open the Terminal and use the `ifconfig` or `ip address` command to determine whether your attacker machine (Kali Linux) is connected to the targeted network (`172.30.1.0/24`), as shown here:

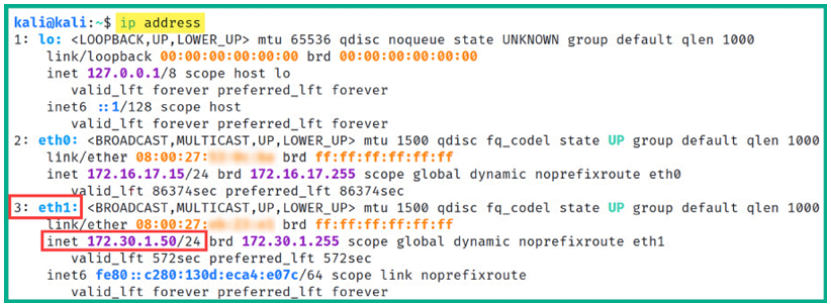


Figure 6.9: Checking your network

As an aspiring ethical hacker and penetration tester, it's important to verify whether your attacker machine has a valid IP address and subnet mask on the targeted network during the internal network penetration test. As shown in

the preceding screenshot, `eth1` is connected to the *PentestNet* environment, which is our targeted network.

Keep in mind that wired network adapters are identified with `eth`, and wireless adapters are identified with `wlan`.

Additionally, the `inet` field contains the IP address that's assigned on the interface of the Kali Linux virtual machine. However, the IP address shown in the preceding screenshot may be different from the address shown on your machine; that's okay once it's on the `172.30.1.0/24` network. Furthermore, identifying the IP address on the network adapter will enable us to exclude scanning our own machine in the next steps.

Ethical hackers and penetration testers often need to determine the network ID and range of IP addresses within a network before performing host discovery on an internal network. While it's recommended to build a solid foundation on networking prior to learning about cybersecurity and penetration testing, the following website is an online subnet calculator that will help you determine the IP ranges and much more: **https://www.calculator.net/ip-subnet-calculator.html**.

3. Next, we can install a command-line tool to help us quickly determine the IP subnet details for a network, use the following command to install `sipcalc`:

```
kali@kali:~$ sudo apt install -y sipcalc
```

4. Next, to calculate the network address, network range, and broadcast address of the `172.30.1.0/24` network, please use the following command:

```
kali@kali:~$ sipcalc 172.30.1.0/24
```

As shown in the following screenshot, `sipcalc` was able to calculate the network range for us:

```
kali@kali:~$ sipcalc 172.30.1.0/24
-[ipv4 : 172.30.1.0/24] - 0

[CIDR]
Host address            - 172.30.1.0
Host address (decimal)  - 2887647488
Host address (hex)      - AC1E0100
Network address         - 172.30.1.0
Network mask            - 255.255.255.0
Network mask (bits)     - 24
Network mask (hex)      - FFFFFF00
Broadcast address       - 172.30.1.255
Cisco wildcard          - 0.0.0.255
Addresses in network    - 256
Network range           - 172.30.1.0 - 172.30.1.255
Usable range            - 172.30.1.1 - 172.30.1.254
```

Figure 6.10: Network range

5. Next, let's use **Netdiscover** to passively scan for live systems on the *PentestNet* environment (`172.30.1.0/24`), using the following command:

```
kali@kali:~$ sudo netdiscover -p -i eth1
```

The `-i` syntax is commonly used to specify the listening interface, and using the `-p` syntax performs a passive scan by enabling Netdiscover to capture and analyze ARP messages on a network by analyzing the source and destination IP and MAC addresses, which helps us to identify live hosts on a network, as shown here:

```
Currently scanning: (passive)   |   Screen View: Unique Hosts

11 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 660
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
_____________________________________________________________________________
172.30.1.1      08:00:27:e9:16:8a      2     120  PCS Systemtechnik GmbH
0.0.0.0         08:00:27:d7:cc:d8      4     240  PCS Systemtechnik GmbH
172.30.1.49     08:00:27:33:ac:4e      3     180  PCS Systemtechnik GmbH
172.30.1.48     08:00:27:d7:cc:d8      2     120  PCS Systemtechnik GmbH
```

Figure 6.11: Live hosts on a network

As shown in the preceding screenshot, Netdiscover provided the IP addresses, MAC addresses, vendors, and hostnames of the live systems on the targeted network. Where `172.30.1.48` is assigned to **Metasploitable 3 – (Windows version)** and `172.30.1.49` is assigned to the **Metasploitable 2** virtual machine. Furthermore, leveraging the MAC vendor information helps us determine the type of devices on the network and can be useful when researching security vulnerabilities for a specific system.

6. Next, to perform an active host discovery scan using Netdiscover, use the following command:

```
kali@kali:~$ sudo netdiscover -r 172.30.1.0/24 -i eth1
```

Since the active scan does not wait for the ARP message, Netdiscover sends its own probes to all usable IP addresses within the `172.30.1.0/24` network. Only live systems will respond, enabling Netdiscover to analyze each response message to identify the IP and MAC addresses of live hosts on the network, as shown here:

```
Currently scanning: Finished!   |   Screen View: Unique Hosts

3 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 180
 IP            At MAC Address     Count     Len  MAC Vendor / Hostname
172.30.1.1      08:00:27:e9:16:8a      1      60  PCS Systemtechnik GmbH
172.30.1.48     08:00:27:d7:cc:d8      1      60  PCS Systemtechnik GmbH
172.30.1.49     08:00:27:33:ac:4e      1      60  PCS Systemtechnik GmbH
```

Figure 6.12: Netdiscover host discovery

To learn more about Netdiscover, please visit https://github.com/netdiscover-scanner/netdiscover.

7. Next, let's use **Network Mapper** (**Nmap**) to perform a *ping sweep* over the entire targeted network and exclude our attacker machine during the scanning process. Use the following commands:

```
kali@kali:~$ nmap -sn 172.30.1.0/24 --exclude 172.30.1.50
```

A **ping sweep** is a basic scanning technique that's used by IT professionals to determine which systems are online within a network. It's the automated process of pinging each usable IP address within a network and observing which devices are responding. However, the `ping` utility within an operating system sends the **Internet Control Message Protocol** (**ICMP**) **ECHO Request** message to the destination and a live system will respond with an **ICMP ECHO Reply** message.

It's a common security practice for cybersecurity professionals to disable ICMP responses on critical systems within their organization. This reduces the likelihood that a novice hacker is to discover a live host. Therefore, if an attacker sends **ICMP ECHO Request** messages to a system that's configured to not respond, the novice attacker will think the target is offline.

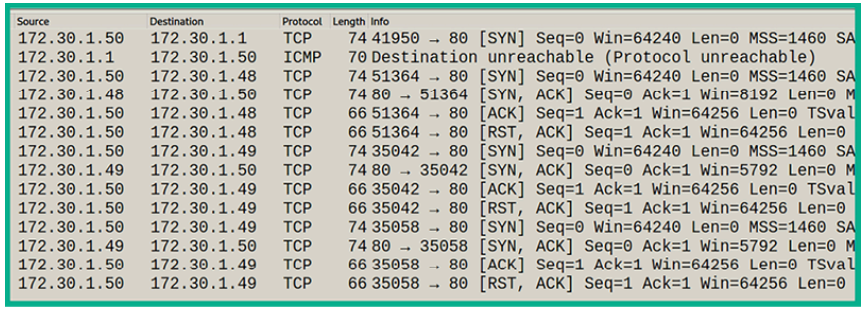
On the other hand, seasoned threat actors and penetration testers who understand the security vulnerabilities that exist within the **Transmission Control Protocol/Internet Protocol** (**TCP/IP**) networking model can bypass this minor security mechanism and instead send TCP messages to specific ports on the targeted system. This technique leverages the design of the TCP and tricks the targeted system into responding, indicating it's live on the network.

The following screenshot shows there are 2 live hosts, `172.30.1.48` and `172.30.1.49`, on the network:

```
kali@kali:~$ nmap -sn 172.30.1.0/24 --exclude 172.30.1.50
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-25 13:29 EDT
Nmap scan report for 172.30.1.48
Host is up (0.00081s latency).
Nmap scan report for 172.30.1.49
Host is up (0.00072s latency).
Nmap done: 255 IP addresses (2 hosts up) scanned in 8.83 seconds
```

Figure 6.13: Ping sweep using Nmap

The `-sn` syntax on Nmap is used to specify a ping scan but Nmap does not send ICMP messages to the target. Instead, Nmap sends TCP messages to specific ports on the targeted system, as shown in the Wireshark packet capture here:



Source	Destination	Protocol	Length	Info
172.30.1.50	172.30.1.1	TCP	74	41950 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SA
172.30.1.1	172.30.1.50	ICMP	70	Destination unreachable (Protocol unreachable)
172.30.1.50	172.30.1.48	TCP	74	51364 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SA
172.30.1.48	172.30.1.50	TCP	74	80 → 51364 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 M
172.30.1.50	172.30.1.48	TCP	66	51364 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval
172.30.1.50	172.30.1.48	TCP	66	51364 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0
172.30.1.50	172.30.1.49	TCP	74	35042 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SA
172.30.1.49	172.30.1.50	TCP	74	80 → 35042 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 M
172.30.1.50	172.30.1.49	TCP	66	35042 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval
172.30.1.50	172.30.1.49	TCP	66	35042 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0
172.30.1.50	172.30.1.49	TCP	74	35058 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SA
172.30.1.49	172.30.1.50	TCP	74	80 → 35058 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 M
172.30.1.50	172.30.1.49	TCP	66	35058 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval
172.30.1.50	172.30.1.49	TCP	66	35058 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0

Figure 6.14: Wireshark packet capture

Nmap sends specially crafted TCP **synchronization** (**SYN**) packets to the targeted host, with the intention of triggering a TCP **reset** (**RST**) or TCP **acknowledgment** (**ACK**) as a response from a live/online host.

To learn more about how the TCP establishes a connection with a destination host using the TCP three-way handshake, please see https://hub.packtpub.com/understanding-network-port-numbers-tcp-udp-and-icmp-on-an-operating-system/.

Identifying live hosts on a network helps ethical hackers and penetration testers create a network topology and identify whether their targets are online before proceeding to profile the targets. Next, you will learn how to identify open ports and running services and determine the operating system of a target.

Identifying open ports, services, and operating systems

After performing host discovery, the next step is to identify any open ports on the targeted system and determine which services are mapped to those open ports. There are various techniques that a penetration tester can use to identify the open ports on a targeted system. Some techniques are manual, while others can simply be automated using the Nmap tool.

To get started fingerprinting using Nmap, please use the following instructions:

1. Firstly, ensure the **Kali Linux**, **Metasploitable 2** and **Metasploitable 3** (**Windows version**) virtual machines are powered on.

2. On **Kali Linux**, open the Terminal and use the following commands to perform a basic Nmap scan to determine whether any of the top 1,000 ports are open on the **Metasploitable 3 (Windows version)** virtual machine:

```
kali@kali:~$ nmap 172.30.1.48
```

As shown in the following screenshot, Nmap indicates there are 20 TCP open ports and provides the name of their associated services:

```
kali@kali:~$ nmap 172.30.1.48
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-25 14:19 EDT
Nmap scan report for 172.30.1.48
Host is up (0.00020s latency).
Not shown: 980 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
3389/tcp  open  ms-wbt-server
4848/tcp  open  appserv-http
7676/tcp  open  imqbrokerd
8009/tcp  open  ajp13
8080/tcp  open  http-proxy
8181/tcp  open  intermapper
8383/tcp  open  m2mservices
```

Figure 6.15: Discovering open ports

Using the information from this scan enables you to start fingerprinting your targeted systems. As a penetration tester, you can determine which ports are open and discover how they can be used as a point of entry into the target and look for security vulnerabilities on the running services.

As an aspiring ethical hacker and penetration tester, it's okay if you don't initially understand the role and function of service ports on a system. However, it is recommended to perform research on anything you're not familiar with to gain a better understanding of the technology or topic. For instance, there are many service ports and each is associated with a specific application-layer service, such as TCP port `443` being associated with

the **Hypertext Transfer Protocol Secure** (**HTTPS**) protocol that's used for secure web communication.

3. Next, let's perform an advanced scan to identify the targeted system's operating system and service versions and retrieve **Server Message Block** (**SMB**) details, using the following command:

```
kali@kali:~$ nmap -A -T4 -p- 172.30.1.48
```

Let's take a look at each syntax that was used in the preceding command:

1. `-A`: This enables Nmap to profile the target to identify its operating system, service versions, and script scanning, as well as perform a traceroute.
2. `-T`: This syntax specifies the timing options for the scan, which ranges from 0–5, where 0 is very slow and 5 is the fastest. This command is useful for preventing too many probes from being sent to the targeted system too quickly, which may trigger alerts.
3. `-p`: Using the `-p` syntax allows you to specify the targeted ports to identify them as open or closed on a system. You can specify `-p80` to scan for port `80` only on the target and `-p-` to scan for all 65,535 open ports.

By default, Nmap scans TCP ports only. Therefore, if a target is running a service on a **User Datagram Protocol** (**UDP**) server port, there's a possibility you will miss it. To perform a UDP scan on a port or range of ports, use the `-p U:53` command, where `53` is the targeted UDP port number.

The following screenshot shows the upper portion of the scan results:

```
kali@kali:~$ nmap -A -T4 -p- 172.30.1.48
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-25 14:39 EDT
Nmap scan report for 172.30.1.48
Host is up (0.00044s latency).
Not shown: 65495 closed tcp ports (conn-refused)
PORT      STATE SERVICE             VERSION
21/tcp    open  ftp                 Microsoft ftpd
| ftp-syst:
|_  SYST: Windows_NT
22/tcp    open  ssh                 OpenSSH 7.1 (protocol 2.0)
| ssh-hostkey:
|   2048 fd:08:98:ca:3c:e8:c1:3c:ea:dd:09:1a:2e:89:a5:1f (RSA)
|_  521 7e:57:81:8e:f6:3c:1d:cf:eb:7d:ba:d1:12:31:b5:a8 (ECDSA)
80/tcp    open  http                Microsoft IIS httpd 7.5
|_http-server-header: Microsoft-IIS/7.5
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Site doesn't have a title (text/html).
135/tcp   open  msrpc               Microsoft Windows RPC
139/tcp   open  netbios-ssn         Microsoft Windows netbios-ssn
445/tcp   open  ◆◆◆-iU              Windows Server 2008 R2 Standard 7601
1617/tcp  open  java-rmi            Java RMI
```

Figure 6.16: Scan result

As shown in the preceding screenshot, Nmap was able to retrieve a lot more in-depth information about our target, such as the service versions of each service that is associated with an open port. It was also able to perform banner grabbing and determine whether there's an authentication system/login mechanism for each service.

The following screenshot is the remaining portion of the same scan results:

```
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 1h00m00s, deviation: 2h38m45s, median: 0s
| smb-os-discovery:
|   OS: Windows Server 2008 R2 Standard 7601 Service Pack 1 (Windows Server 2008 R2 Standard 6.1)
|   OS CPE: cpe:/o:microsoft:windows_server_2008::sp1
|   Computer name: vagrant-2008R2
|   NetBIOS computer name: VAGRANT-2008R2\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2023-08-25T11:44:07-07:00
```

Figure 6.17: Operating system profiling

As shown in the preceding screenshot, Nmap was able to identify the host operating system on the target as a **Windows Server 2008 R2** machine with **Service Pack 1**. In addition, Nmap was able to determine the hostname of the system and whether it's connected to a domain or not based on the workgroup name.

Whenever a Windows-based system is not connected to a **domain controller** (**DC**), the default Workgroup is called `Workgroup`. Furthermore, the Nmap scan was able to perform a basic SMB scan to identify the operation system, which also indicates that the targeted system may have file and printer shares available.

The following are additional syntaxes that can be used during the scanning process with Nmap:

- `-Pn`: This syntax enables Nmap to perform a scan on the targeted systems without first performing host discovery, and simply considers the target to be online.
- `-sU`: This syntax enables Nmap to perform UDP port scanning on the targeted systems. This command will be useful in identifying whether there are any running services on UDP ports as compared to TCP port numbers.
- `-p`: This syntax allows you to specify either a range of targeted ports or specific ports that are open on a system. Using `nmap -p 50-60`, `nmap -p 80,443`, or `nmap -p 22` allows you to scan a range, a group, or specific port numbers. However, using `nmap -p-` specifies to scanning all 65,535 port numbers, but keep in mind that Nmap scans TCP ports by default.
- `-sV`: This syntax enables you to perform service version identification of running services on a targeted system. For instance, an Nmap basic scan may indicate port `23` is open and associated with the **Telnet**. As an ethical hacker, it is important to determine the service version of this running service. Therefore, using the `nmap -sV <targeted system>` command will identify the service version, which can be useful when researching security vulnerabilities on a target.
- `-6`: Using this syntax enables Nmap to perform scans on a targeted IPv6 network or a host with an IPv6 address.

Additionally, ethical hackers and penetration testers can use the `ping` utility to profile the operating system of a target by analyzing the **time-to-live** (**TTL**) value found within the ICMP response messages from the target. For instance, Windows-based operating systems reply with a default TTL value of `128`, while Linux-based systems reply with a default TTL value of `64`.

To learn more about the TTL value within an IP packet, please visit **https://www.techtarget.com/searchnetworking/definition/time-to-live**.

To better understand how ICMP helps us identify the operating system of a targeted machine, please use the following instructions:

1. On **Kali Linux**, use the following commands to send **4 ICMP ECHO** Request messages to the **Metasploitable 3 (Windows version)** virtual machine:

```
kali@kali:~$ ping 172.30.1.48 -c 4
```

As shown in the following screenshot, all ICMP responses contain a TTL of `128`, which indicates the targeted system is running a version of the Windows operating system:

```
kali@kali:~$ ping 172.30.1.48 -c 4
PING 172.30.1.48 (172.30.1.48) 56(84) bytes of data.
64 bytes from 172.30.1.48: icmp_seq=1 ttl=128 time=0.514 ms
64 bytes from 172.30.1.48: icmp_seq=2 ttl=128 time=0.260 ms
64 bytes from 172.30.1.48: icmp_seq=3 ttl=128 time=0.314 ms
64 bytes from 172.30.1.48: icmp_seq=4 ttl=128 time=0.307 ms

--- 172.30.1.48 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3059ms
rtt min/avg/max/mdev = 0.260/0.348/0.514/0.097 ms
```

Figure 6.18: Sending ICMP ECHO Request messages to Metasploitable 3

2. Next, use the following commands to send 4 ICMP ECHO Request messages to the **Metasploitable 2** virtual machine:

```
kali@kali:~$ ping 172.30.1.49 -c 4
```

As shown in the following screenshot, the ICMP responses have a TTL value of `64`, which indicates the targeted system is running a version of Linux:

```
kali@kali:~$ ping 172.30.1.49 -c 4
PING 172.30.1.49 (172.30.1.49) 56(84) bytes of data.
64 bytes from 172.30.1.49: icmp_seq=1 ttl=64 time=0.226 ms
64 bytes from 172.30.1.49: icmp_seq=2 ttl=64 time=0.269 ms
64 bytes from 172.30.1.49: icmp_seq=3 ttl=64 time=0.214 ms
64 bytes from 172.30.1.49: icmp_seq=4 ttl=64 time=0.238 ms

--- 172.30.1.49 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3057ms
rtt min/avg/max/mdev = 0.214/0.236/0.269/0.020 ms
```

Figure 6.19: Sending ICMP ECHO Request messages to Metasploitable 2

As an aspiring ethical hacker and penetration tester, identifying the operating system, open ports, and running services helps you to better profile the target and identify its security vulnerabilities. By identifying the security vulnerabilities, you can improve your exploit development phase and plan of attack. Simply put, an exploit or payload for a Windows-based operating system will most likely not work on a Linux-based system, or vice versa. However, it's worth noting that while exploit development is an advanced skill, many penetration testers utilize existing exploits, adapting their approach based on the target's vulnerabilities.

Thus far, you have learned how to discover open ports, service versions, operating systems, and SMB versions. Next, you will learn how to evade detection while performing active scanning on a network and systems using Nmap.

Using scanning evasion techniques

Whenever a packet is sent from one device to another, the source and destination IP addresses are included within the header of the packet. This is the default behavior of the TCP/IP networking model; all addressing information must be included within all packets before they are placed on the network. When performing a scan as an ethical hacker and a penetration tester, we try to remain undetected to determine whether the security team of the targeted organization has the capabilities of detecting the simulated cyberattack.

During a real cyberattack, if an organization is unable to detect suspicious activities and security incidents on their network and systems, the threat actor can simply achieve their objectives without obstructions. However, if an organization can detect suspicious activities as soon as they occur, the security team can take action quickly to contain and stop the threat while safeguarding their organization's

assets. During a penetration test, it's important to simulate real-world cyberattacks to test the threat detection and mitigation systems within the targeted organization.

Avoiding detection with decoys

Nmap is usually considered to be the king of network scanners within the cybersecurity industry due to its advanced scanning capabilities like operating system identification, service version detection, and scriptable interactions with the targeted system through the **Nmap Scripting Engine** (**NSE**). Nmap enables penetration testers to use decoys when scanning a targeted system. This scanning technique tricks the targeted system into thinking the source of the scan is originating from multiple sources, rather than a single-source IP address that belongs to the attacker machine.

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

1. Power on the **Kali Linux**, **Metasploitable 2**, and **Metasploitable 3 (Windows version)** virtual machines. Kali Linux will be the attacker machine, **Metasploitable 2** will be the targeted system, and the **Metasploitable 3 (Windows version)** virtual machine will be the decoy, as shown in the following diagram:

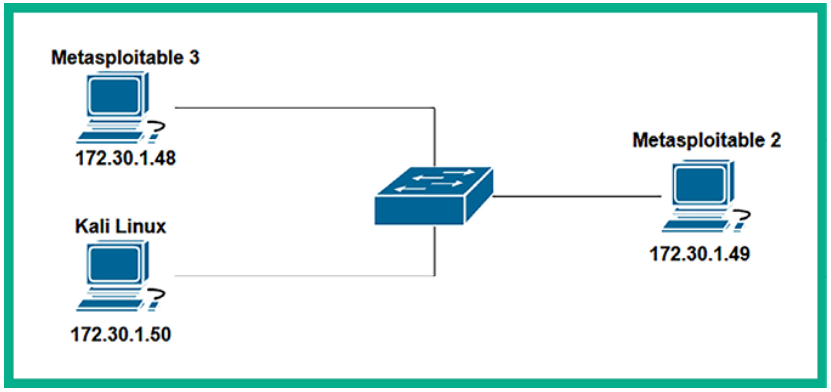


Figure 6.20: Decoys

Ensure that you identify the IP addresses of each of these systems as they may be different from the preceding diagram. Using the scanning techniques from the previous section will help you identify the IP addresses easily.

2. Next, to perform an Nmap scan using decoys, use the following command:

```
kali@kali:~$ sudo nmap 172.30.1.49 -D 172.30.1.48
```

Using the `-D` syntax enables you to specify one or more decoys. Before Nmap uses the decoy addresses, it will first check whether each decoy system is a live host on the network and whether an address is reachable; it won't include the off-line address during the scan.

The following screenshot shows the expected results of the scan:

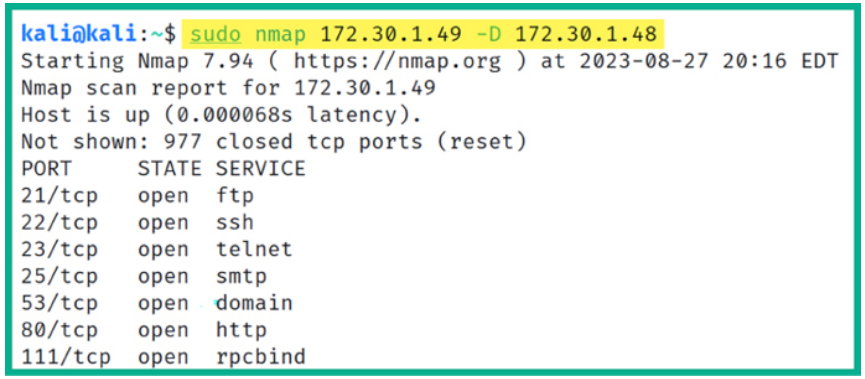


Figure 6.21: Specifying decoys

If the security team of the targeted organization is closely monitoring the packets over their internal network and identifies that a port scan is in progress, there's a chance they will determine that the scan originates from your IP address. However, the decoy feature will include the decoy addresses within various packets from your attacker machine, as shown here:

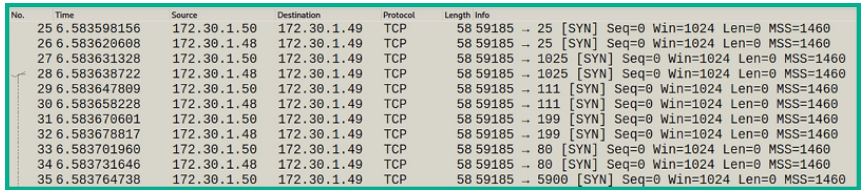


Figure 6.22: Decoy addresses

Therefore, using more decoy addresses during the Nmap scan will decrease the risk of a security analyst tracing the source of the scan back to your IP address. However, security analysts are well-trained professionals and usually have the required tools and skills to identify threats quickly on their network infrastructure.

Using MAC and IP spoofing techniques

Nmap is like the Swiss Army knife of scanners, filled with lots of scanning features to evade detection. Nmap allows a penetration tester to spoof both the MAC and IP addresses of their Kali Linux machine.

The following are common MAC and IP spoofing techniques with Nmap:

1. To perform an Nmap scan using a randomized MAC address, use the `--spoof-mac 0` command as shown here:

```
kali@kali:~$ sudo nmap --spoof-mac 0 172.30.1.49
```

The following screenshot shows that Nmap generated a random MAC address before performing the scan on the targeted system:

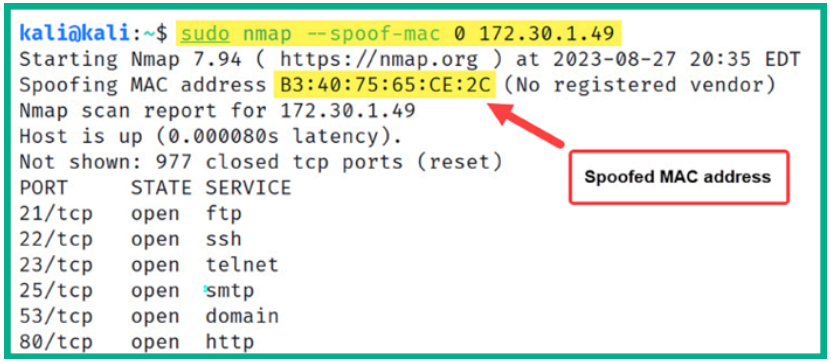


Figure 6.23: Spoofed MAC address

In addition, the following screenshot shows the packets that were captured using Wireshark to further verify that Nmap used a randomized address as the source MAC address:

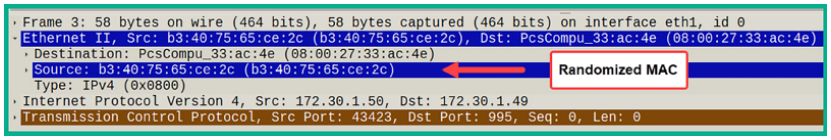


Figure 6.24: Verifying spoofed MAC address

2. Performing an Nmap scan on a targeted system with a spoof MAC address of a specific vendor is as simple as including the vendor's name, with the following command:

```
kali@kali:~$ sudo nmap -sT -Pn --spoof-mac hp 172.30.1.49
```

The following screenshot shows Nmap using an HP MAC address as the source address:

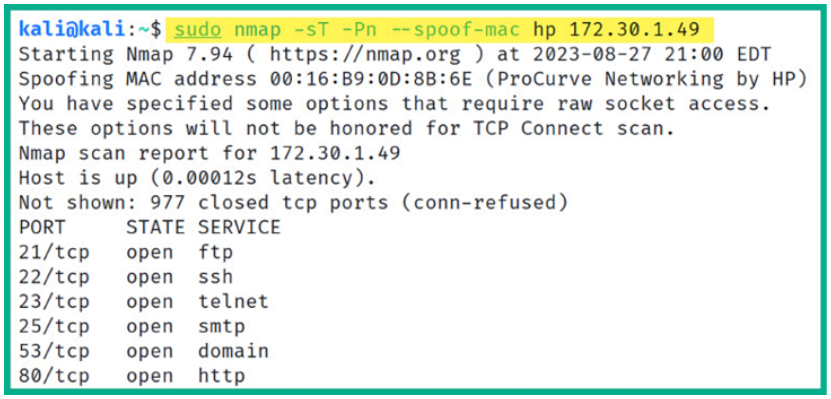


Figure 6.25: Spoofed MAC address of a specific vendor

To learn more about the various functionalities of Nmap, use the `nmap -h` and `man nmap` commands to view the help menu and manual page, respectively.

Having completed this section, you have learned how to evade detection on a network while performing scanning using Nmap. Next, you will learn how to perform a stealth scan using Nmap.

Stealth scanning techniques

By default, Nmap establishes a TCP three-way handshake on any open TCP ports found on the targeted systems. Once the handshake has been established between the attacker machine and the targeted system, data packets are exchanged between each host.

The following diagram shows the TCP 3-way handshake, where Host A is initializing communication with Host B:

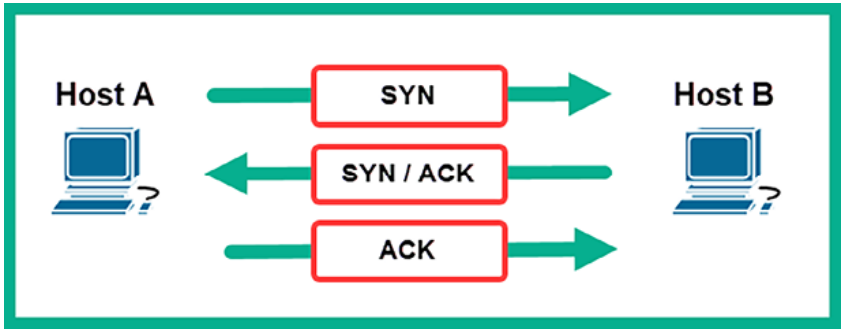


Figure 6.26: TCP 3-way handshake

During a penetration test, it's important to be as stealthy as possible on the network. This creates the effect of a real adversary attempting to compromise the targeted systems on the network, without being caught by the organization's security solutions. However, by establishing a TCP three-way handshake with the targeted devices, we are making ourselves known to the target.

By using Nmap, we can perform a stealth scan (half-open/**SYN** scan) between the target and our attacker system. A stealth scan does not set up a full TCP three-way handshake, but resets the connection before it is fully established.

With a stealth scan, the sender initiates a TCP connection by sending a **SYN** packet, but does not complete the three-way handshake as it sends an **RST** packet instead of an **ACK** packet after receiving a **SYN/ACK** from the target. **SYN**, **ACK**, **RST**, and **FIN** are TCP flags found within a TCP packet to indicate the state of the connection between a source and destination device.

The following diagram shows the exchange of TCP packets during an Nmap stealth scan:

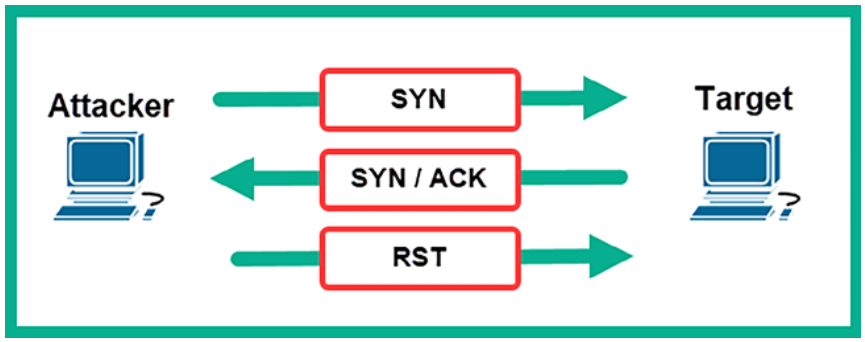


Figure 6.27: Stealth scan

Let's break down what is shown in the preceding diagram:

1. The attacker machine tricks the target by sending a **TCP SYN** packet to a specific port on the targeted system to determine whether the port is open.
2. Then, the target system will respond with a **TCP SYN/ACK** packet if the port is open.
3. Lastly, the attacker will send a **TCP RST** packet to the target to reset and terminate the connection.

To get started with learning stealth scanning techniques, please use the following instructions:

1. Power on the **Kali Linux** and **Metasploitable 2** virtual machines.
2. On **Kali Linux**, open the Terminal and use the following commands to perform a stealth scan on **Metasploitable 2** to identify whether port `80` is open:

```
kali@kali:~$ sudo nmap -sS -p 80 172.30.1.48
```

   Using the `-sS` syntax to indicate a stealth scan and the `-p` operator allows us to specify a target port.
   The following screenshot shows the Nmap identified port `80` as open on the targeted system:

```
kali@kali:~$ sudo nmap -sS -p 80 172.30.1.48
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-28 19:19 EDT
Nmap scan report for 172.30.1.48
Host is up (0.00017s latency).

PORT   STATE SERVICE
80/tcp open  http
MAC Address: 08:00:27:D7:CC:D8 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 6.79 seconds
```

Figure 6.28: Stealth scanning using Nmap

The following screenshot shows the exchange of packets between **Kali Linux** (`172.30.1.50`) and the targeted system (`172.30.1.48`) during the stealth scan:

Source	Destination	Protocol	Length	Info
172.30.1.50	172.30.1.48	TCP	58	49795 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
172.30.1.48	172.30.1.50	TCP	60	80 → 49795 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
172.30.1.50	172.30.1.48	TCP	54	49795 → 80 [RST] Seq=1 Win=0 Len=0

Figure 6.29: Wireshark capture

As shown in the preceding snippet, Nmap sends a **TCP SYN** packet with a destination port `80` to identify whether there are any running services on port `80` of the targeted system. The target responded with a **TCP SYN/ACK** packet as expected, however, the attacker machine sent a **TCP RST** packet to reset and terminate the connection. Therefore, no network connections were made between the attacker machine (**Kali Linux**) and the targeted system during the stealth scan.

However, keep in mind that seasoned cybersecurity professionals who are actively monitoring their network traffic for any security incidents can easily identify whether a threat actor is performing a stealth scan on their network.

Additionally, you can consider performing ACK scans, Window scans, Fragmentation scans, Idle scans, and UDP scanning. These techniques can be found using the `man nmap` command on Kali Linux.

Having completed this section, you have learned how to perform various types of scanning techniques to identify live hosts on a network and profile their running

services and operating systems. In the next section, you will learn how to enumerate common services and network shares from vulnerable systems.

Enumerating network services

While scanning, you will notice that there are common network services running on the targeted systems. Collecting more information on these network services can help you further identify shared network resources such as shared directories, printers, and file shares on the system.

Sometimes, these network services are misconfigured and enable a threat actor to gain unauthorized access to sensitive data stored on servers and other systems within an organization. By performing enumeration on network services running a targeted system, we'll be able to identify user accounts, network shares, and password policies, and profile the target's operation system. Using the information collected during enumeration helps us to better understand which security vulnerabilities exist and how to improve our plan of attack on the target.

Over the next few subsections, you will learn how to enumerate common network services such as SMB, the **Simple Mail Transfer Protocol (SMTP)**, and the **Simple Network Management Protocol (SNMP)**.

Enumerating SMB services

After identifying vulnerabilities and misconfigurations through comprehensive scanning, the next step is the detailed enumeration of specific network services, a process that may reveal potential entry points for attackers.

SMB is a common network service that allows hosts to share resources, such as files, with other devices on a network. As an aspiring ethical hacker and penetration tester, it's always recommended to enumerate file shares once it's within your scope for the penetration test.

To get started enumerating SMB services on a targeted system, please use the following instructions:

1. Power on both the **Kali Linux** and **Metasploitable 2** virtual machines.
2. On **Kali Linux**, open the Terminal and use the following command to launch the Metasploit framework:

```
kali@kali:~$ msfconsole
```

3. After the Metasploit framework loads, use the `search` command along with the `smb_version` search term to quickly locate modules:

```
msf6 > search smb_version
```

As shown in the following screenshot, the search result shows only one module available, which can be used to identify whether SMB is running on the targeted system and its version:

Figure 6.30: Searching for modules

4. Next, use the following commands to load the module and display its options:

```
msf6 > use auxiliary/scanner/smb/smb_version
msf6 auxiliary(scanner/smb/smb_version) > options
```

Using the `options` or `show options` command displays the current settings within the loaded module and helps you determine whether there are additional configurations needed before executing the module, as shown here:

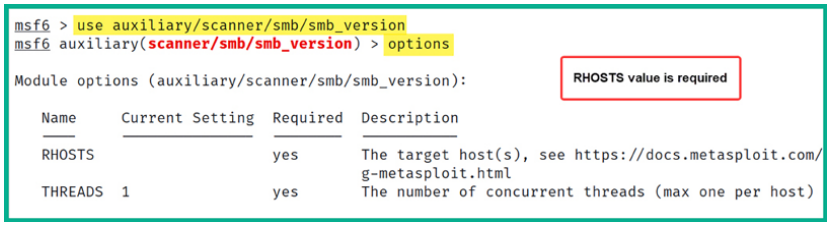


Figure 6.31: Viewing module options

As shown in the preceding screenshot, there are two required settings. One is `RHOSTS` or the target settings, while the other is the number of threads to apply to the process. Notice that the `RHOSTS` setting is blank.

5. Next, use the following commands to set the targeted system (**Metasploitable 2**) as `RHOSTS` and execute the module:

```
msf6 auxiliary(scanner/smb/smb_version) > set RHOSTS 172.30.1.49
msf6 auxiliary(scanner/smb/smb_version) > run
```

The `run` command is commonly used to execute auxiliary modules within the Metasploit framework, while the `exploit` command is used to execute exploit modules.

The following screenshot shows that Metasploit was able to detect that SMB is running and its version from the targeted system:

```
msf6 auxiliary(scanner/smb/smb_version) > set RHOSTS 172.30.1.49
RHOSTS => 172.30.1.49
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 172.30.1.49:445       - SMB Detected (versions:1) (preferred dialect:) (signatures:optional)
[*] 172.30.1.49:445       -   Host could not be identified: Unix (Samba 3.0.20-Debian)
[*] 172.30.1.49:          - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 6.32: Enumerating SMB

Use the `exit` command to quit the Metasploit framework and return to the Bash shell on the Terminal.

Using more than one tool to enumerate running services on your target is always recommended because there's a possibility that one tool may not identify something important. Sometimes, penetration testers may prefer to work with Metasploit as it contains a lot of *auxiliary* modules to scan and enumerate services, while others prefer Nmap. However, I recommend that you become familiar with both tools as they are excellent and will be very handy in various situations.

Since SMB has been discovered on the targeted system, we can use **SMBMap** to enumerate the files and shared drives within the target.

To get started using SMBMap, please use the following instructions:

1. Ensure that the **Kali Linux** and **Metasploitable 2** virtual machines are powered on.
2. On **Kali Linux**, use the following commands on the Terminal to identify whether the targeted system (**Metasploitable 2**) is running the SMB service:

```
kali@kali:~$ nmap -p 139,445 172.30.1.49
```

3. The following screenshot shows that Nmap was able to identify ports `139` and `445` as open on the targeted system:

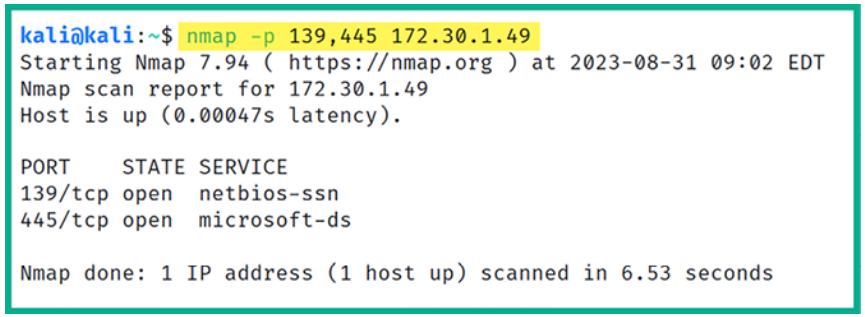


Figure 6.33: Using Nmap

4. Next, use SMBMap to identify whether the targeted system has any network shares:

```
kali@kali:~$ smbmap -H 172.30.1.49
```

As shown in the following screenshot, the targeted system (**Metasploitable 2**) has a few shared drives, but most are not accessible over the network except for the `tmp` resource:

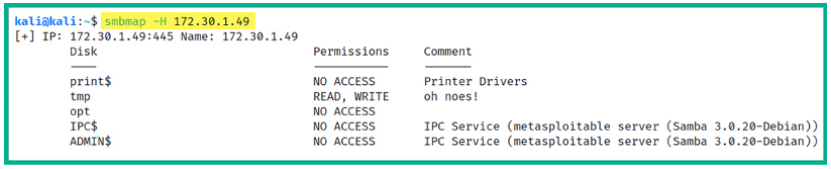


Figure 6.34: Discovering shared drives

As shown in the preceding screenshot, the SMBMap tool was able to provide the permissions and comments for each network share on a targeted system. This information is useful in helping ethical hackers and penetration testers identify sensitive directories and collect data found within unsecure network shares.

5. Next, use the following commands to display the contents of the `tmp` directory on the targeted system:

```
kali@kali:~$ smbmap -H 172.30.1.49 -r tmp
```

As shown in the following screenshot, SMBMap was able to access the `tmp` directory because there was no authentication mechanism configured to restrict unauthenticated access:

```
kali@kali:~$ smbmap -H 172.30.1.49 -r tmp
[+] IP: 172.30.1.49:445 Name: 172.30.1.49
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        tmp                                                     READ, WRITE
        .\tmp\*
        dr--r--r--                0 Mon Aug 28 20:11:47 2023    .
        dw--w--w--                0 Sun May 20 14:36:11 2012    ..
        fw--w--w--                0 Mon Aug 28 18:49:42 2023    4582.jsvc_up
        dr--r--r--                0 Mon Aug 28 18:49:31 2023    .ICE-unix
        dr--r--r--                0 Mon Aug 28 18:49:36 2023    .X11-unix
        fw--w--w--               11 Mon Aug 28 18:49:36 2023    .X0-lock
```

Figure 6.35: Displaying contents of a shared drive

6. Next, to download all the contents of the `tmp` directory onto your Kali Linux machine, use the following commands to create a new directory (folder) within Kali Linux and download the files:

```
kali@kali:~$ mkdir smb_files
kali@kali:~$ cd smb_files
kali@kali:~/smb_files$ smbmap -H 172.30.1.49 --download .\tmp\*
```

The following screenshot shows the execution of the preceding commands:

```
kali@kali:~$ mkdir smb_files
kali@kali:~$ cd smb_files
kali@kali:~/smb_files$ smbmap -H 172.30.1.49 --download .\tmp\*
```

Figure 6.36: Creating a new directory and downloading files

An additional tool that's already pre-installed within Kali Linux is `enum4linux`, which enables ethical hackers and penetration testers to perform system enumeration on a targeted system. This tool retrieves the usernames, password policies, SMB shares, and operating system information of a targeted system. To learn more about `enum4linux`, please visit https://www.kali.org/tools/enum4linux/.

Having completed this section, you have learned how to perform SMB enumeration using both Metasploit and SMBMap. In the next section, you will learn how to perform SMTP enumeration.

Enumerating SMTP services

Enumerating SMTP services enables ethical hackers and penetration testers to collect information about email services and identify any valid user accounts on the targeted system.

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

1. Power on both the **Kali Linux** and **Metasploitable 2** virtual machines.
2. On **Kali Linux**, open the Terminal and use `netcat` to check whether port `25` is open on the targeted system (**Metasploitable 2**) and identify the running service:

```
kali@kali:~$ nc -nv 172.30.1.49 25
```

3. Next, use the `VRFY root` command to determine whether `root` is a valid user.
4. Next, use `VRFY toor` to check whether the `toor` user is a valid user, as shown here:

```
kali@kali:~$ nc -nv 172.30.1.49 25
(UNKNOWN) [172.30.1.49] 25 (smtp) open
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
VRFY root
252 2.0.0 root
VFTY toor
502 5.5.2 Error: command not recognized
```

Figure 6.37: Checking whether port is open

As shown in the preceding screenshot, `netcat` is able to successfully establish a connection to the targeted system on port `25`, which further identifies that the SMTP is running. When the `VRFY root` command is executed, the email service responses indicate that the user exists. However, the email service provides an error message when a non-valid user is checked.

When performing SMTP enumeration, there are various commands that enable us to verify whether a valid user exists or not. For instance, the `VRFY` command is used to determine whether a valid user exists on the email server. The `EXPN` command is used to identify the delivery address for an email alias. The `RCPT TO` command is used to point to a recipient's email address.

5. Manually checking each possible username on a targeted system can be very time-consuming. To help automate the process of SMTP enumeration, we can use a simple BASH script that intakes a pre-defined list of possible usernames and queries it on the targeted system.
   To download the script onto your Kali Linux machine, use the following command:

```
kali@kali:~$ wget https://raw.githubusercontent.com/PacktPublishing/The-Ultimate-Kali-Linux-Book-3E/main/Chapter%2006/
```

6. Next, use the following command to view the contents of the script:

```
kali@kali:~$ cat smtp_user_enum.sh
```

Once the preceding command executes, the following code is shown:

```
#!/bin/bash
if [ $# -ne 2 ]; then
    echo "Usage: $0 <target_ip> <email_list>"
    exit 1
fi
target_ip="$1"
email_list="$2"
echo "Starting SMTP user enumeration..."
while IFS= read -r email; do
    # Construct the SMTP communication
    ( sleep 1; echo "HELO example.com"; sleep 1; echo "VRFY $email"; sleep 1; echo "QUIT" ) | nc -nv $target_ip 25 | g
    if [ $? -eq 0 ]; then
        echo "User found: $email"
    fi
done < "$email_list"
echo "SMTP user enumeration finished."
```

7. Next, use the following commands to make the newly saved script executable on Kali Linux:

```
kali@kali:~$ chmod +x smtp_user_enum.sh
```

8. To use the script, the `./smtp_user_enum.sh <target> <wordlist>` syntax enables you to start the SMTP enumeration on a targeted system, as with the following command:

```
kali@kali:~$ ./smtp_user_enum.sh 172.30.1.49 /usr/share/wordlists/seclists/SecLists-master/Usernames/top-usernames-sho
```

The following screenshot shows that valid usernames are identified while the script is running:

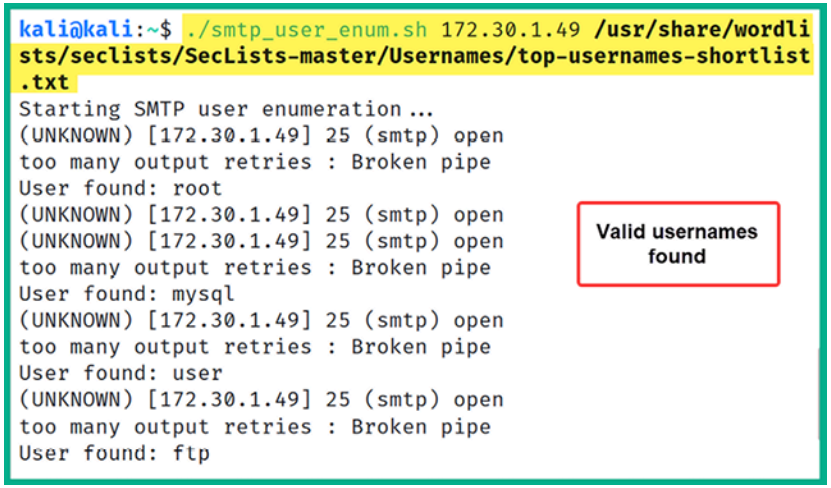


Figure 6.38: Finding valid usernames

Identifying and leveraging valid usernames and accounts helps penetration testers gain unauthorized access to targeted systems. However, it's important to remember that performing such activities should only be conducted within a legal framework, such as part of an agreed-upon penetration testing contract where written permission has been explicitly granted by the system owner. Having completed this exercise, you have gained hands-on skills in SMTP enumeration. Next, you will learn how to enumerate SNMP services on a targeted host.

Enumerating SNMP services

SNMP is a common network protocol that enables network professionals to monitor, manage, and troubleshoot common networking devices. In addition, IT professionals use SNMP to retrieve sensitive information from their devices, such as the following:

- System uptime
- Device hostname
- CPU and memory utilization
- Interface status and statistics
- Operating system
- Open ports and running services

SNMP leverages the **management information base** (**MIB**), which is a common database that contains specific information about an SNMP-managed device. The MIB is like a tree structure that's divided into multiple branches and each branch is used to manage a specific area of the device. On each branch of the MIB tree, there are leaves that represent specific values that enable a network professional to access the leaves to retrieve specific information about the device on the network.

To learn more about SNMP, please see https://www.techtarget.com/searchnetworking/definition/SNMP. To learn specifically about the different versions of SNMP, please visit https://www.splunk.com/en_us/blog/learn/snmp-monitoring.html.

To get started with SNMP enumeration, please use the following instructions:

1. Power on the **Kali Linux** and **Metasploitable 3 (Windows version)** virtual machines.
2. On **Kali Linux**, open the Terminal and use the following command to determine whether SNMP is running on the targeted system (**Metasploitable 2**):

```
kali@kali:~$ sudo nmap -sU -p 161 172.30.1.48
```

The following screenshot shows SNMP is running on the targeted system on UDP port `161`:

```
kali@kali:~$ sudo nmap -sU -p 161 172.30.1.48
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-31 10:28 EDT
Nmap scan report for 172.30.1.48
Host is up (0.00028s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 08:00:27:D7:CC:D8 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 6.71 seconds
```

Figure 6.39: Checking if SNMP is running on the targeted system

3. Next, perform SNMP enumeration using the **SNMP-Check** tool, use the following command:

```
kali@kali:~$ snmp-check -p 161 -c public -v 1 172.30.1.48
```

The following is a description of each syntax used in the preceding command:

1. `-p` : This allows you to specify the targeted port; by default, it's set to port `161` .
2. `-c` : This allows you to specify the community string to log in to the targeted system; the default community string is `public` .
3. `-v` : This allows you to specify the SNMP version to use; by default, it's set to version `1` .

As shown in the following screenshot, we are able to identify a lot of sensitive information that can be used to improve future cyberattacks on the target:

```
kali@kali:~$ snmp-check -p 161 -c public -v 1 172.30.1.48
snmp-check v1.9 - SNMP enumerator
Copyright (c) 2005-2015 by Matteo Cantoni (www.nothink.org)

[+] Try to connect to 172.30.1.48:161 using SNMPv1 and community 'public'

[*] System information:

  Host IP address               : 172.30.1.48
  Hostname                      : vagrant-2008R2
  Description                   : Hardware: AMD64 Family 25 Model 80 Stepping
.1 (Build 7601 Multiprocessor Free)
  Contact                       : -
  Location                      : -
  Uptime snmp                   : 00:05:04.28
  Uptime system                 : 00:04:48.41
  System date                   : 2023-8-31 07:33:28.6
  Domain                        : WORKGROUP

[*] User accounts:

  sshd
  Guest
  greedo
  vagrant
```

Figure 6.40: SNMP enumeration

The SNMP-Check tool was able to enumerate the following information from the target:

- System information
- User accounts
- Network information
- Routing information
- Network services
- Running processes
- Software components

To learn more about SNMP-Check, use the `snmp-check -h` command to display its menu and additional options.

As you have learned, enumerating systems helps ethical hackers and penetration testers improve their profiles on targeted systems and determine what's running on them. Such information helps penetration testers identify vulnerabilities that can be exploited to compromise the target.

In the next section, you will learn how to discover data leaks in cloud storage.

Discovering data leaks in the cloud

Over the past decade, cloud computing has become one of the fastest-growing trends in the IT industry. Cloud computing allows companies to migrate and utilize computing resources within a cloud provider's data center. Cloud computing providers have a pay-as-you-go model, which means that you only pay for the resources you use. Some cloud providers allow pay-per-minute models, while others use a pay-per-hour structure.

The following are popular cloud computing service providers and the storage services provided by them:

- **Amazon Web Services (AWS)**: The AWS storage facility is known as **Simple Storage Service** (**S3**). Whenever a customer enables the S3 service, a bucket is created. A bucket is a storage unit within the AWS platform where the customer can add or remove files.
- **Microsoft Azure**: In Microsoft Azure, the file storage facility is known as Azure Files.
- **Google Cloud Platform**: On Google Cloud, the storage facility is known as Google Cloud Storage.

- **Oracle Cloud Infrastructure (OCI)**: On OCI, the storage facility is known as Oracle Cloud Infrastructure Object Storage.

In the field of cybersecurity, we need to remember that when a company is using a cloud platform, the data on the cloud platform must be secured, just like it should be when stored on-premises (that is, when stored locally). Sometimes, administrators forget to enable security configurations or lack knowledge regarding the security of a cloud solution. This could lead to, say, an attacker discovering a target organization's AWS S3 buckets and downloading their content.

For this exercise, we are going to use some free online learning resources from `http://flaws.cloud`. This is a learning environment that's been created by an AWS security professional who is helping the community learn about security vulnerabilities that can exist within AWS S3 misconfigurations.

To get started with identifying data leakage with AWS S3 buckets, please use the following instructions:

1. Power on the **Kali Linux** virtual machine, open the Terminal, and use the following commands to install the **S3Scanner** tool:

```
kali@kali:~$ sudo apt update
kali@kali:~$ sudo pip3 install s3scanner
```

2. Next, install the AWS command-line package using the following command:

```
kali@kali:~$ sudo apt install awscli
```

3. Next, configure the AWS command-line features on Kali Linux by using the following command:

```
kali@kali:~$ aws configure
```

   Simply hit **Enter** to use the default options, as shown in the following screenshot:

```
kali@kali:~$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]:
Default output format [None]:
```

Figure 6.41: Configuring AWS command-line features

4. Next, to view all the supported features and options of the S3Scanner tool, use the `s3scanner -h` command, as shown in the following screenshot:

```
kali@kali:~$ s3scanner -h
usage: s3scanner [-h] [--version] [--threads n] [--endpoint-url ENDPOINT_URL] [--endpoint-address-style {path,vhost}]
                 [--insecure]
                 {scan,dump} ...

s3scanner: Audit unsecured S3 buckets
           by Dan Salmon - github.com/sa7mon, @bltjetpack

options:
  -h, --help            show this help message and exit
  --version             Display the current version of this tool
  --threads n, -t n     Number of threads to use. Default: 4
  --endpoint-url ENDPOINT_URL, -u ENDPOINT_URL
                        URL of S3-compliant API. Default: https://s3.amazonaws.com
  --endpoint-address-style {path,vhost}, -s {path,vhost}
                        Address style to use for the endpoint. Default: path
  --insecure, -i        Do not verify SSL

mode:
  {scan,dump}           (Must choose one)
    scan                Scan bucket permissions
    dump                Dump the contents of buckets
```

Figure 6.42: S3scanner tool

5. Next, let's use **NsLookup** within Kali Linux to retrieve the IP address of the targeted server:

```
kali@kali:~$ nslookup flaws.cloud
```

The following screenshot shows that NsLookup was able to retrieve multiple public IP addresses for the hosting server:

```
kali@kali:~$ nslookup flaws.cloud
Server:         172.16.17.18
Address:        172.16.17.18#53

Non-authoritative answer:
Name:   flaws.cloud
Address: 52.92.148.75
Name:   flaws.cloud
Address: 52.92.227.27
Name:   flaws.cloud
Address: 52.218.182.154
```

Figure 6.43: Retrieving public IP addresses

6. Next, let's use NsLookup again to retrieve the hostname of the AWS S3 bucket server:

```
kali@kali:~$ nslookup 52.92.148.75
```

The following screenshot shows the hostname of the server, including the name of the AWS S3 bucket:

```
kali@kali:~$ nslookup 52.92.148.75
75.148.92.52.in-addr.arpa       name = s3-website-us-west-2.amazonaws.com.
```

Figure 6.44: Showing hostname

An AWS S3 bucket's URL format is usually in the form of `https://<bucketname>.s3. <region>.amazonaws.com`. Therefore, by using the information from the URL, the following can be determined:

1. S3 bucket name: `s3-website`
2. Hosting region: `us-west-2`

AWS S3 buckets are not only used to store data such as files. They are also used to host websites. Therefore, we can use `flaws.cloud` as a prefix to the AWS S3 bucket URL to get the following URL: **http://flaws.cloud.s3-website-us-west-2.amazonaws.com**

The following screenshot shows the contents of the preceding URL:

Figure 6.45: Viewing URL content

While the website in this exercise does not use HTTPS, it's recommended to always use HTTPS for security reasons in a real-world scenario as it provides data encryption.

7. Next, let's use S3Scanner to verify that a bucket exists and the available permissions:

```
kali@kali:~$ s3scanner scan --bucket http://flaws.cloud
```

As shown in the following screenshot, an AWS S3 bucket exists:

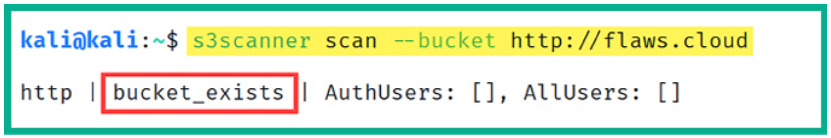


Figure 6.46: Verifying whether a bucket exists

8. Next, let's attempt to view the contents of the AWS S3 bucket using the following command:

```
kali@kali:~$ aws s3 ls s3://flaws.cloud --region us-west-2 --no-sign-request
```

As shown in the following screenshot, there are some files within the S3 bucket:

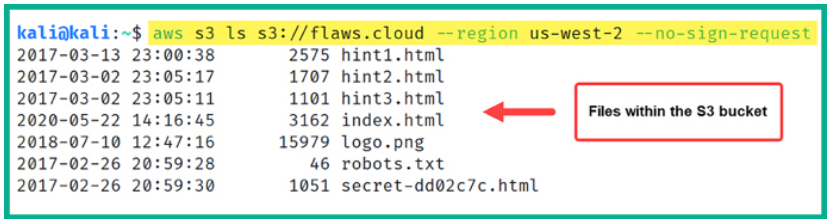


Figure 6.47: Viewing bucket content

9. Next, let's attempt to download the files onto our Kali Linux machine. Use the following commands to create a folder and download the files into the newly created folder:

```
kali@kali:~$ mkdir s3_bucket_files
kali@kali:~$ cd s3_bucket_files
kali@kali:~/s3_bucket_files$ aws s3 cp s3://flaws.cloud/secret-dd02c7c.html --region us-west-2 --no-sign-request secre
```

The `cp` syntax specifies the file to download, `--region` allows us to specify the location of the AWS S3 bucket, and `--no-sign-request` specifies us to not use any user credentials.

10. Lastly, you can use the `cat` or `open` command to view the contents of the downloaded file, as shown here:

```
kali@kali:~/s3_bucket_files$ cat secret-dd02c7c.html
kali@kali:~/s3_bucket_files$ open secret-dd02c7c.html
```

You can continue this exercise on **http://flaws.cloud/** to learn more about various security vulnerabilities and discover the impact of misconfigurations on cloud services such as AWS S3 buckets. However, do not perform such actions on systems, networks, and organizations that you do not have legal permission to do so.

As you have seen, data leaks can happen on any platform and to any organization. As an aspiring ethical hacker and penetration tester, you must know how to find them before a real adversary does and exploits them. Companies can store sensitive data on cloud platforms, or even leave data completely unprotected on a cloud service provider network. This can lead to data and accounts being re-

trieved. In this section, you learned how to perform enumeration of AWS S3 buckets using various tools and techniques.

Summary

In this chapter, you have gained hands-on skills as an aspiring ethical hacker and penetration tester to perform active scanning techniques to identify open ports, running services, and operating systems on targeted systems. In addition, you have learned how to use common evasion techniques during scanning to reduce your threat level. Furthermore, you have discovered how to enumerate common network services and leverage the information to improve a cyberattack.

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 Vulnerability Assessments*, you will learn how to set up and work with popular vulnerability management tools.

Further reading

- Nmap reference guide – **https://nmap.org/book/man.html**
- Information gathering using Metasploit – **https://www.offensive-security.com/metasploit-unleashed/information-gathering/**

Join our community on Discord

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

https://packt.link/SecNet